Convergence Analysis of Generalized Primal-Dual Interior-Point Algorithms for Linear Optimization

by

Hua Wei

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2002

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

We study the zeroth-, first-, and second-order algorithms proposed by Tunçel. The zeroth-order algorithms are the generalization of the classic primal-dual affine-scaling methods, and have a strong connection with the quasi-Newton method. Although the zeroth-order algorithms have the property of strict monotone decrease in both primal and dual objective values, they may not converge. We give an illustrative example as well as an algebraic proof to show that the zeroth-order algorithms do not converge to an optimal solution in some cases. The second-order algorithms use the gradients and Hessians of the barrier functions. Tunçel has shown that all second-order algorithms have a polynomial iteration bound. The second-order algorithms have a range of primal-dual scaling matrices to be chosen. We give a method to construct such a primal-dual scaling matrix. We then analyze a new centrality measure. This centrality measure appeared in both first- and second-order algorithms. We compare the neighbourhood defined by this centrality measure with other known neighbourhoods. We then analyze how this centrality measure changes in the next iteration in terms of the step length and some other information of the current iteration.

Acknowledgements

I would like to express my deep thanks to my supervisor, Dr. Levent Tunçel. Without his continues guidance and support, I could not finish this thesis. I would also like to thank the two readers, Dr. Michael Best and Dr. Henry Wolkowicz, for their comments and careful reading of the draft.

Thanks to the professors, colleagues, and friends in the Department of Combinatorics and Optimization at the University of Waterloo.

Thanks my parents, my brother for their love and continues encouragement. Although they were not in Canada when I was writing the thesis, I can always feel their support.

Last, I owe great thanks to my wife, Feng Zou, for her love, encouragement, and being my company in my life for countless good or bad days.

Contents

List of Figures

Chapter 1

Introduction

The linear programming problem is an optimization problem with a linear objective function and linear constraints. Dantzig [2] [3] proposed the simplex method to solve linear programming problems. The simplex method is very efficient. With the development of computer technology, the simplex method has been successfully implemented on computers. Even now, most commercial linear programming solvers still use the simplex method.

Although the simplex method is very efficient on average, no variant of it has been proven to have polynomial complexity in the worst case. Klee and Minty [13] devised a problem such that the simplex method (employing the largest coefficient rule) needs exponential time to solve in terms of the problem size (data length). The exponential worst-case complexity of the simplex method motivated many researchers to look for a polynomial time algorithm for the linear programming problem.

Khachiyan [11], [12] first found a polynomial time algorithm for linear programming problem by using the ellipsoidal method of Shor [28], and Yudin and Nemirovskii [41]. Khachiyan's result has a great impact in theory, but in practice, the algorithm always achieves the worst-case bound, and can not beat the algorithms based on the simplex method.

Karmarkar's seminal paper [10] in 1984 gave a polynomial time algorithm and it was announced as more efficient than the simplex method. Karmarkar's algorithm is very different from the simplex method. The simplex method is a combinatorial method. It iterates through the extreme points of the feasible region to achieve an optimal extreme point. However, Karmarkar's algorithm is more like an algorithm working on a non-linear optimization problem. It evolves through a series of strictly feasible points (interior-points), and approaches an optimal solution.

That is why it and its following variants are called interior-point methods.

Karmarkar's paper led many researchers into this area. Soon, Vanderbei, Meketon, and Freedman [36] and Barnes [1] proposed a natural simplification of Karmarkar's algorithm. It turned out that as early as 1967, Dikin [4] had a very similar proposal. Nowadays, it is called the affine-scaling method. Then, in 1990, Monteiro, Adler, and Resende [22] described a polynomial-time primal-dual affine-scaling algorithm.

Renegar [27] was the first to prove an $O(\sqrt{n}L)$ iteration bound for a path-following method. The path-following methods explicitly restrict the iterates to a neighbourhood of the central path (defined later) and follow the central path to an optimal solution. Although Renegar's path-following method has a better iteration bound than Karmarkar's algorithm, it was not efficient in practice. Later, Vaidya [34] refined Renegar's path-following algorithm, and described an algorithm with an overall complexity of $O(n^3L)$. Vaidya was also the first to extend path-following methods to handle exponential many constraints in some cases as the ellipsoidal method does. Almost at the same time, Gonzaga [6] also proposed an algorithm with the same overall complexity. Kojima, Mizuno, and Yoshise [16] also proposed a primal-dual path-following method at that time. This primal-dual path-following method was soon be improved to the same complexity as Vaidya's ($O(n^3L)$) by these authors themselves [15], and Monteiro and Adler [21].

Another family of algorithms based on the potential-function were developed later. These algorithms use a potential-function to merit each iterate. The potential-function keeps the iterates away from the boundary and rewards the improvements in the objective function at the same time. At each iteration, the algorithms try to decrease the potential function by a constant, and in return, the algorithms get a polynomial-iteration bound.

Karmarkar's algorithm is a primal-only potential-reduction algorithm. Gonzaga [7] gave a primal-only potential-reduction algorithm. His algorithm and Karmarkar's algorithm are both $O(nL)$-iteration algorithms.

In 1990, Todd and Ye [31] gave the first algorithm using the primal-dual potential function in its analysis. But their algorithm was more like a path-following algorithm than a potential-reduction algorithm. The first pure potential-reduction algorithm with $O(\sqrt{n}L)$-iteration complexity bound was due to Ye [39] (see also Freund [5]). This algorithm was not a symmetric primal-dual method. In the same year, Kojima, Mizuno and Yoshise [17] gave a symmetric primal-dual potential reduction algorithm with $O(\sqrt{n}L)$ iteration complexity bound.

We would like to refer to some good references here. Gonzaga [8] gave an excellent review on path-following methods. Later, Todd [30] gave an excellent survey paper on potential-function.

Also, there are some recently published books in interior-point method. They are Nesterov and Nemirovskii [24], Vanderbei [35], Wright [38], and Ye [40].

Among all variants of interior-point methods, symmetric primal-dual interior-point methods play a very important role both in theory and in practice. On the one hand, it has been well analyzed in theory and has the best worst-case complexity bound, on the other hand, it is very robust and efficient in practice.

Recently, there is a trend to generalize these already successful algorithms for linear programming to a much wider area, such as semidefinite programming and convex programming. Nesterov and Todd [25] gave a symmetric primal-dual interior-point algorithm for feasible regions expressed as the intersection of a symmetric cone with an affine subspace. Later, Tunçel [33] generalized the primal-dual interior-point methods to convex optimization problem in conic form. His generalization not only generalizes the algorithm's applicable problems, namely, generalizes from linear programming to convex optimization problem in conic form, but also generalizes the search direction to a wider range.

During the last ten years, there has been significant interest in a particular convex optimization problem in conic form: semidefinite programming. Primal-dual interior-point algorithms for such problems have been proposed by Helmberg, Rendl, Vanderbei, Wolkowicz [9], Kojima, Shindoh, Hara [18], Monteiro and Zhang [23]. For an overview of the area, see the handbook [37] edited by Wolkowicz, Saigal, and Vandenberghe.

We will do some research based on the generalized algorithms. We limit our discussion to linear programming problems, and focus on the related problems arisen from the generalization of the search direction. Tunçel [33] provides three frameworks for primal-dual algorithms, namely *Zeroth-, First-, and Second-Order Algorithms.* The difference of the algorithms lies in how the barrier function is used.

- zeroth-order algorithms need no information about the barrier functions (the primal barrier function and the dual barrier function), but need feasible region information. In return, descent in the primal and dual objective function values can be proved.

- first-order algorithms need the first derivatives of the barrier functions. In return, the first-order descent in the potential function can be proved.

- second-order algorithms need both the gradients and the Hessians of the barrier functions.

In return, conditions for constant reduction in the potential function can be obtained, and hence an $O(\sqrt{n}\ln(1/\epsilon))$-iteration complexity bound is obtained.

The zeroth-order algorithms have a strong connection with the quasi-Newton methods by using a BFGS like update to construct the primal-dual scaling matrix. While the second-order algorithms are more like the approach of Nesterov and Nemirovskii [24], and Nesterov and Todd [25], [26].

This thesis has been structured in five Chapters. The first one is the introduction, which is what we are going through now. The second chapter presents the fundamental theorems of linear programming which is necessary for the theorems developed in the later chapters. The third chapter analyzes the zeroth-order algorithm, and points out why the algorithm fails to converge in some cases using an illustrative example. An algebraic proof for the non-convergence of the zeroth-order algorithm in some cases is also presented in this chapter. The fourth chapter gives an introduction for the first- and the second-order algorithms, and then analyzes the primal dual scaling matrix in the second-order algorithm. The fifth chapter discusses the proximity measures for the central path.

Chapter 2

Fundamentals of Linear Programming

2.1 Notations

Here is a list of notations for reference. We will also mention the notation when we use it.

$\mathbb{R}^n$, $\mathbb{R}^n_+$:	the set of n-dimensional real vectors, the set of non-negative vectors in $\mathbb{R}^n$;
$\mathcal{F}(P)$:	the feasible set of the primal problem;
$\mathcal{F}_+(P)$:	the strictly feasible set of the primal problem;
$\mathcal{F}(D)$:	the feasible set of the dual problem;
$\mathcal{F}_+(D)$:	the strictly feasible set of the dual problem;
x:	the primal variable;
y, s:	the dual variable;
$\hat{x}, \hat{y}, \hat{s}$:	some specific variables;
$\bar{x}, \bar{y}, \bar{s}$:	the variables in the scaled space;
$\mathcal{N}(A)$:	the null space of A;
$\mathcal{R}(A)$:	the range of A, or column space of A;
P_A	$:= I - A^T(AA^T)^{-1}A$. The orthogonal projection matrix onto $\mathcal{N}(A)$;
Σ^n:	the set of symmetric $n \times n$ real matrices;
Σ^n_+:	the set of symmetric positive semidefinite $n \times n$ real matrices;
Σ^n_{++}:	the set of symmetric positive definite $n \times n$ real matrices;

e $:= [1, 1, \ldots, 1]^T$;
X: the diagonal matrix such that $Xe = x$;
$\lambda_i(A)$: the i^{th} largest eigenvalue of a matrix A;
$\lambda(A)$: the vector consisting of the eigenvalues of a symmetric matrix A;
$\text{Diag}(v)$: a diagonal matrix such that $\text{Diag}(v)e = v$.

2.2 Basic Theorems of Linear Programming

We consider the linear programming (LP) problems in the following form:

$$\begin{array}{lrcl} (P) & \min c^T x \,, & & \\ & Ax & = & b \,, \\ & x & \geq & 0 \,, \end{array}$$

where c is in $\mathbb{R}^n$, A is in $\mathbb{R}^{m\times n}$, and b is in $\mathbb{R}^m$. We use $x \geq 0$ to denote that every element of the vector x is greater than or equal to 0. Similarly, we use $x > 0$ to denote that every element of vector x is greater than 0.

Without loss of generality, we assume A has full row rank. If A does not have full row rank, then either the linear system $Ax = b$ has no solution, or $Ax = b$ has some redundant equations. We can remove those redundant equations without changing the solution set of $Ax = b$, and then have a new matrix which is of full row rank.

We use (P) to denote the primal problem, the corresponding dual problem is written in the following form:

$$\begin{array}{lrcl} (D) & \min b^T y \,, & & \\ & A^T y + s & = & c \,, \\ & s & \geq & 0 \,, \end{array}$$

where A, b, and c are the same data discussed in the primal problem (P). Therefore, y is in $\mathbb{R}^m$, and s is a slack variable in $\mathbb{R}^n_+$.

We use the following definitions to denote the feasible region:

$$\begin{array}{rcl} \mathcal{F}(P) & := & \{x \in \mathbb{R}^n_+ : Ax = b\} \,, \\ \mathcal{F}_+(P) & := & \{x \in \mathbb{R}^n_{++} : Ax = b\} \,, \\ \mathcal{F}(D) & := & \{y \in \mathbb{R}^m, \ s \in \mathbb{R}^n_+ : A^T y + s = c\} \,, \\ \mathcal{F}_+(D) & := & \{y \in \mathbb{R}^m, \ s \in \mathbb{R}^n_{++} : A^T y + s = c\} \,. \end{array}$$

We also say a variable is primal or dual feasible if it is in the set $\mathcal{F}(P)$ or $\mathcal{F}(D)$ respectively; a variable is strictly primal or dual feasible if it is in the set $\mathcal{F}_+(P)$ or $\mathcal{F}_+(D)$ respectively. We assumed that A is full row rank. Also, we assume that $\mathcal{F}_+(P)$ and $\mathcal{F}_+(D)$ are not empty. So we can start the interior-point method from a strictly feasible point.

It seems that we have two variables (y, s) in the dual problem. We call s the dual slack variable. But in another view, we show that y is uniquely determined by s. If $(\hat{y}, \hat{s})$ is a feasible solution of the dual problem (D), because $A^T\hat{y} = c - \hat{s}$, and A is full row rank, then $\hat{y}$ is uniquely determined by $\hat{s}$. For convenience, sometimes we will only write s to refer to the dual feasible solution.

With the assumption that A has full row rank, we can define

$$P_A := I - A^T(AA^T)^{-1}A \; . \tag{2.1}$$

This matrix is known as "the orthogonal projection matrix onto $\mathcal{N}(A)$, the null space of A". That is, for any $x \in \mathbb{R}^n$, P_Ax is in the null space of A. This can be verified by seeing that $A(P_Ax) = Ax - AA^T(AA^T)^{-1}Ax = 0$.

We need the following fundamental theorem of linear algebra: (we use $\perp$ to denote the orthogonal complement to a linear sub-space.)

Lemma 2.1 $\mathcal{N}(A)^\perp =$ *row space of* A $=$ *column space of* $A^T = \mathcal{R}(A^T)$.

Using the above lemma, we can write the dual problem in another form, which only uses the variable s.

Theorem 2.2 *Let $\hat{x}$ be any feasible solution to the primal, then the dual problem (D) is equivalent to the following problem (D'),*

$$\begin{aligned} (D') \qquad & \max -\hat{x}^T s, \\ & P_A s = P_A c, \\ & s \geq 0. \end{aligned}$$

Proof
We use $\mathcal{F}(D') := \{s \in \mathbb{R}^n_+ : P_As = P_Ac\}$ to denote the feasible region of (D'). For any $(\hat{y}, \hat{s}) \in \mathcal{F}(D)$, we have $A^T\hat{y} = c - \hat{s}$, so $(c - \hat{s})$ is in the $\mathcal{R}(A^T)$. Because $\mathcal{R}(A^T)$ is the orthogonal complement of $\mathcal{N}(A)$, we have $P_A(c - \hat{s}) = 0$. Thus $\hat{s}$ is in $\mathcal{F}(D')$. Conversely, if $\hat{s}$ is in

$\mathcal{F}(D')$, then $P_A(c-\hat{s})=0$ means $(c-\hat{s})$ is in the range of A^T. Thus there exists a $\hat{y}$, such that $A^T\hat{y}+\hat{s}=c,\ \hat{s}\geq 0$. So $\mathcal{F}(D')=\mathcal{F}(D)$. Notice that $b^Ty=\hat{x}^TA^Ty=\hat{x}^T(c-s)=-\hat{x}^Ts+\hat{x}^Tc$. This means the difference between the objective value of (D) and (D') is just a constant $\hat{x}^Tc$. Because they have the same feasible region, and the difference between the objective functions is just a constant, these two problems are equivalent. □

The following is the well-known weak duality relation.

Fact 2.3 (Weak duality relation) *Let $\hat{x}$ and $(\hat{y},\hat{s})$ be a feasible solution for (P) and (D) respectively, then the primal objective value is greater than or equal to the dual objective value, that is*

$$c^T\hat{x}\geq b^T\hat{y},\quad \textit{and}\ \ c^T\hat{x}-b^T\hat{y}=\hat{x}^T\hat{s}\ .$$

Proof

$$c^T\hat{x}=(A^T\hat{y}+\hat{s})^T\hat{x}=\hat{y}^TA^Tx+\hat{s}^T\hat{x}=\hat{y}^Tb+\hat{x}^T\hat{s}\ .$$

Because $\hat{x}\geq 0$ and $\hat{s}^T\geq 0$, we have $c^T\hat{x}\geq b^T\hat{y}$. □

Theorem 2.4 (Strong duality theorem) *Let $\hat{x}$ and $(\hat{y},\hat{s})$ be feasible solutions for (P) and (D) respectively. Then the optimal solutions for (P) and (D) exist, and the optimal values for (P) and (D) are equal.*

Directly using the weak duality relation and the strong duality theorem, we see that x and s are optimal solutions to the primal and dual problems if and only if they satisfy the following system:

$$\begin{aligned} Ax &= b, \\ A^Ty+s &= c, \\ x^Ts &= 0, \\ x,s &\geq 0. \end{aligned} \tag{2.2}$$

2.3 Central Path

We define a pair of families of non-linear programming problems, parameterized by $\mu \geq 0$:

$$\begin{array}{lrcl} (P_\mu) & \min c^T x - \mu \sum_{i=1}^{n} \log x_i \ , & & \\ & Ax & = & b \ , \\ & (x & > & 0) \ . \\ (D_\mu) & \min -b^T y - \mu \sum_{i=1}^{n} \log s_i \ , & & \\ & A^T y + s & = & c \ , \\ & (s & > & 0) \ . \end{array}$$

Here, and throughout this thesis, all logarithms are base-2. The functions $-\sum_{i=1}^{n} \log x_i$ and $-\sum_{i=1}^{n} \log s_i$ are called the barrier functions (for primal and dual respectively). These barrier functions give a description of the inequality constraints. So, the inequality constraints are implicit here.

Theorem 2.5 *Suppose the primal and the dual problems both have strictly feasible solutions, then (P_μ) and (D_μ) have unique solution pair $x(\mu)$, $(y(\mu), s(\mu))$ for each $\mu > 0$.*

Proof
Note that the objective function $c^T x - \mu \sum_{i=1}^{n} \log x_i$ is strictly convex. And when x_i goes to zero, the objective value goes to infinity. So the minimizer exists and it must be an interior point. The objective function is strictly convex, so the minimizer must be unique.

Similarly, we can prove that the dual solution is also unique. □

Theorem 2.6 *Suppose the primal and the dual problems both have strictly feasible solutions. Then for a fixed $\mu > 0$, the unique solution $x(\mu)$, $(y(\mu),\ s(\mu))$ of (P_μ) and (D_μ) make up the unique solution to the following system:*

$$\begin{array}{rcl} Ax & = & b, \quad x > 0 \ , \\ A^T y + s & = & c, \quad s > 0 \ , \\ Xs & = & \mu e \ . \end{array} \tag{2.3}$$

Proof

We use Karush-Kuhn-Tucker (KKT) condition to prove the theorem. For the parameterized primal problem (P_μ), the Lagrangian function and its derivatives are:

$$\begin{aligned} L(x,\lambda) &:= (c^Tx - \mu\sum_{i=1}^{n}\log x_i) - (Ax-b)^T\lambda \ , \\ \nabla_x L(x,\lambda) &= c - \mu X^{-1}e - A^T\lambda \ , \\ \nabla^2_{xx} L(x,\lambda) &= X^{-2} \ . \end{aligned}$$

The Hessian of the Lagrangian is positive definite. So, the KKT condition, which is $\nabla_x L(x,\lambda) = 0$, is both sufficient and necessary in our case. Let $s := \mu X^{-1}e > 0$, $y := \lambda$, then $Xs = \mu e$. Moreover, $\nabla_x L(x,\lambda) = 0$ is equivalent to $A^Ty + s = c$. Also, because x is a feasible solution to the problem (P_μ), we must have $Ax = b$ and $x > 0$. Thus system (2.3) is a restatement of the KKT condition of problem (P_μ). So, a solution of system (2.3) is equivalent to the optimal solution of (P_μ). Theorem 2.5 shows that (P_μ) has a unique solution. Thus, this also proves that the solution of system (2.3) is unique by Theorem 2.5.

To prove that the parameterized dual problem (D_μ)'s solution is also a solution of the system (2.3), we use the different form of the dual problem from Theorem 2.2. We can rewrite the problem (D_μ) in the following equivalent form:

$$\begin{aligned} (D_\mu{}') \qquad & \min \ \hat{x}^Ts - \mu\sum_{i=1}^{n}\log s_i \ , \\ & P_Ac = P_As \ , \\ & (s > 0) \ , \end{aligned}$$

where $\hat{x}$ is any feasible solution to the primal problem. The Lagrangian for the problem $(D_\mu{}')$ and its derivatives are:

$$\begin{aligned} L(s,\lambda) &:= (\hat{x}^Ts - \mu\sum_{i=1}^{n}\log s_i) - [P_A(c-s)]^T\lambda \ , \\ \nabla_s L(s,\lambda) &= \hat{x} - \mu S^{-1}e + P_A^T\lambda = 0 \ , \\ \nabla^2_{ss} L(s,\lambda) &= S^{-2} \ . \end{aligned}$$

Similarly, the KKT condition is also sufficient and necessary condition here. Let $x := \mu S^{-1}e > 0$, then $Xs = \mu e$. Also,

$$\begin{aligned} \nabla_s L(s,\lambda) = 0 \ &\Rightarrow \ A\nabla_s L(s,\lambda) = A\bar{x} - Ax + 0 = 0 \\ &\Rightarrow \ Ax = b \ . \end{aligned}$$

Moreover, s is a feasible solution of (D_μ'), which is just (D_μ), so $A^T y + s = c$. Hence, the optimal solution of (D_μ') must also be a solution to the system (2.3). □

If a feasible solution pair (x, s) satisfy system (2.3) for some $\mu > 0$, then we say they are on the central path.

As μ goes to 0, $x(\mu)^T s(\mu)$, which is μn, also goes to 0. So if $x(\mu)$ and $s(\mu)$ converge, then $x(\mu)$ and $s(\mu)$ must converge to a solution of the system (2.2), which is an optimal solution pair to the primal (P) and dual (D) problem. McLinden [20] proved the following theorem for the monotone linear complementarity problem, which includes linear programming.

Theorem 2.7 *Let $(x(\mu), y(\mu), s(\mu))$ be on the central path. Then $(x(\mu), s(\mu))$ converges to an optimal solution pair for primal (P) and dual (D) problem.*

So, if we can find a solution pair of (P_μ) and (D_μ), and decrease μ at each iteration, we will achieve an optimal solution. This is the basic idea behind the path-following methods. Since it is expensive to get an exact optimal solution for (P_μ) and (D_μ), usually, we find an approximate solution near the optimal solution (the central path), and then decrease μ, go to the next iteration. To measure the approximation, a neighbourhood of the central path is defined.

The following are some of the neighbourhoods of the central path.

Example 1: $\mathcal{N}_2(\beta) := \{(x, s) \in \mathcal{F}_+(P) \oplus \mathcal{F}_+(D) : \|Xs - \mu e\|_2 \leq \beta\mu\}$.

Example 2: $\mathcal{N}_\infty(\beta) := \{(x, s) \in \mathcal{F}_+(P) \oplus \mathcal{F}_+(D) : \|Xs - \mu e\|_\infty \leq \beta\mu\}$.

Example 3: $\mathcal{N}_\infty^-(\beta) := \{(x, s) \in \mathcal{F}_+(P) \oplus \mathcal{F}_+(D) : \|Xs - \mu e\|_\infty^- \leq \beta\mu\}$.

Here, for $v \in \mathbb{R}^n$, $\|v\|_\infty^- := -\min\{0, \min_j\{v_j\}\}$.

Clearly, for $v \in \mathbb{R}^n$, $\|v\|_2 \geq \|v\|_\infty \geq \|v\|_\infty^-$. So, for every $\beta \geq 0$, we have

$$\mathcal{N}_2(\beta) \subseteq \mathcal{N}_\infty(\beta) \subseteq \mathcal{N}_\infty^-(\beta).$$

We also have some other measures of proximity to the central path. We define

$$\psi(x, s) := n \log \frac{x^T s}{n} - \sum_{i=1}^{n} \log x_i - \sum_{i=1}^{n} \log s_i. \tag{2.4}$$

Proposition 2.8 *For all* $x \in \mathbb{R}^n_{++}$, $s \in \mathbb{R}^n_{++}$,

$$\psi(x, s) \geq 0 \, .$$

Moreover, the inequality above holds as equality if and only if $s = tX^{-1}e$ *for some* $t > 0$.

Proof
Using the Arithmetic-Geometric Mean Inequality, we have

$$\begin{aligned} & \frac{\sum_{i=1}^n a_i}{n} \geq \sqrt[n]{\prod_{i=1}^n a_i} \quad (a_i \geq 0) && (2.5) \\ \iff \quad & \frac{\sum_{i=1}^n x_i s_i}{n} \geq \sqrt[n]{\prod_{i=1}^n x_i s_i} \quad (\text{ let } \ a_i =: x_i s_i) \\ \iff \quad & \log \frac{\sum_{i=1}^n x_i s_i}{n} \geq \frac{1}{n} (\sum_{i=1}^n \log x_i + \sum_{i=1}^n \log s_i) \\ \iff \quad & \psi(x, s) \geq 0. \end{aligned}$$

The equality holds in (2.5) if and only if all a_i in (2.5) are equal. Therefore, $\psi(x, s) = 0$ if and only if $s = tX^{-1}e$ for some $t > 0$. □

We define μ as always $x^T s / n$. Following [33], we also define the shadow iterates corresponding to x, s. $\tilde{s} := X^{-1}e$, $\tilde{x} := S^{-1}e$, $\tilde{\mu} := \frac{\tilde{x}^T \tilde{s}}{n}$. First, we have the well-known Arithmetic-Harmonic Mean Inequality.

Lemma 2.9 *Assume* $a_i \geq 0$ *for all* $i = 1, 2, \ldots, n$. *Then*

$$\sum_{i=1}^n a_i \sum_{i=1}^n \frac{1}{a_i} \geq n^2 \, .$$

The above inequality holds as equality if and only if $a_1 = a_2 = \ldots = a_n$.

Proof
If we expand the left hand side, we have a term $\frac{a_i}{a_j} + \frac{a_j}{a_i}$ for each $i \neq j$. We have $\binom{n}{2}$ such terms. Also for each index i, we have a term $a_i \cdot \frac{1}{a_i} = 1$ in the left hand side. We have n such terms. So, the left hand side $= \sum_{i \neq j} (\frac{a_i}{a_j} + \frac{a_j}{a_i}) + n \geq \binom{n}{2} \cdot 2 + n = n^2$. Note that the inequality holds as equality if and only if $a_i = a_j$ for all i and j. □

Using the last lemma, we conclude a special case of a result of Nesterov and Todd [26] (see also Tunçel [33]).

Lemma 2.10 *For every (x, s) which is a strictly primal and dual feasible solution pair,*

$$\mu\tilde{\mu} \geq 1,$$

and the inequality above holds as equality if and only if $s = \mu X^{-1}e$.

Proof
Note that $\mu\tilde{\mu} \geq 1 \iff (\sum_{i=1}^n x_i s_i)(\sum_{i=1}^n \frac{1}{x_i s_i}) \geq n^2$. Let $a_i := x_i s_i$. Using above lemma (the Arithmetic-Harmonic Mean Inequality), we have the desired result. □

2.4 Primal-dual Potential Function

The following primal-dual potential function was proposed by Tanabe [29], and Todd and Ye [31].

DEFINITION 2.11 (Tanabe-Todd-Ye potential function)

$$\phi_\rho(x, s) := (n + \rho)\log x^T s - \sum_{i=1}^n \log x_i - \sum_{i=1}^n \log s_i, \ \textit{where } \rho \geq 0.$$

Notice that $\phi_\rho(x, s) = \psi(x, s) + \rho \log x^T s$.

We have the following well-known theorem about this potential function and the corresponding proximity measure $\psi(x, s)$:

Theorem 2.12 *Suppose we are given $x^{(0)}$, $s^{(0)}$ strictly feasible in (P) and (D) respectively. Further assume $\psi(x^{(0)}, s^{(0)}) \leq \rho \log(\frac{1}{\epsilon})$ for some given $\epsilon \in (0, 1)$. If we reduce the value of $\phi_\rho(x, s)$ by an absolute positive constant in every iteration, then there exists a $\bar{k} = O(\rho \log(\frac{1}{\epsilon}))$, such that*

$$x^{(k)^T} s^{(k)} \leq \epsilon x^{(0)^T} s^{(0)}, \quad \textit{for all } k \geq \bar{k}.$$

Proof
Let δ denote the absolute positive constant decrease attained in every iteration. Then we have

$$-k\delta \geq \phi_\rho(x^{(k)}, s^{(k)}) - \phi_\rho(x^{(0)}, s^{(0)})$$

$$
\begin{aligned}
&= \rho \log \frac{x^{(k)^T} s^{(k)}}{x^{(0)^T} s^{(0)}} + \psi(x^{(k)}, s^{(k)}) - \psi(x^{(0)}, s^{(0)}) \\
&\geq \rho \log \frac{x^{(k)^T} s^{(k)}}{x^{(0)^T} s^{(0)}} - \log(\frac{1}{\epsilon}).
\end{aligned}
$$

We have used $\psi(x^{(k)}, s^{(k)}) \geq 0,\ \psi(x^{(0)}, s^{(0)}) \leq \rho \log(\frac{1}{\epsilon})$ at the last step. So

$$
\rho \log \frac{x^{(k)^T} s^{(k)}}{x^{(0)^T} s^{(0)}} \leq \rho \log(\frac{1}{\epsilon}) - k\delta.
$$

We choose $\bar{k} := \frac{2}{\delta} \rho \log(\frac{1}{\epsilon})$, then for all $k \geq \bar{k}$ we have

$$
x^{(k)^T} s^{(k)} \leq \epsilon x^{(0)^T} s^{(0)}.
$$

$\square$

Chapter 3

The Zeroth-Order Algorithms and Their Convergence Issue

3.1 The Classical Primal-Dual Affine Scaling Algorithm

Let T be a diagonal positive definite matrix, then a scaling transformation on problem (P) is a change of variable from x to $\bar{x} := T^{-1}x$. A scaling transformation on the dual problem is a change of variable from s to $\bar{s} := Ts$. So we have the new scaled primal problem:

$$
\begin{aligned}
(\bar{P}) \quad \min\ \bar{c}^T\bar{x}\,, & \\
\bar{A}\bar{x} &= b\,, \\
\bar{x} &\geq 0\,,
\end{aligned}
$$

as well as the scaled dual problem

$$
\begin{aligned}
(\bar{D}) \quad \max\ b^T y, & \\
\bar{A}^T y + \bar{s} &= \bar{c}\,, \\
\bar{s} &\geq 0\,,
\end{aligned}
$$

where $\bar{A} := AT$, $\bar{c} := Tc$. The scaled problem $(\bar{P})$ has the same feasible region as the original problem (P), namely for any feasible solution $\hat{x}$ to the original problem, the corresponding scaled solution $T^{-1}\hat{x}$ is also a feasible solution to the scaled primal problem $(\bar{P})$; and vice versa. Also, their objective values are the same. The same argument can be applied on the dual problem. We consider the pair of problems $(\bar{P})$, $(\bar{D})$ equivalent to the pair (P), (D). We call this property *Scale Invariance* (see Tunçel [32]).

The reason why we use scaling is not only because scaling gives us a very clear and beautiful presentation, but also because that after scaling, the steepest descent direction will change, and this may give us a better direction.

The Newton direction to solve the system (2.2) is

$$\begin{aligned} Ad_x &= 0\,, \\ A^T d_y + d_s &= 0\,, \\ Sd_x + Xd_s &= -Xs\,. \end{aligned}$$

If we let $T = X^{1/2}S^{-1/2}$, and scale the original problem (P) and (D), then the above system becomes

$$\begin{aligned} \bar{A}\bar{d_x} &= 0\,, \\ \bar{A}^T d_y + \bar{d_s} &= 0\,, \\ \bar{d_x} + \bar{d_s} &= -v\,, \end{aligned} \tag{3.1}$$

where $\bar{A} := AT$, $\bar{d_x} := T^{-1}d_x$, $\bar{d_s} := Td_s$, and $v := Ts = T^{-1}x$.

Lemma 3.1 *Assume $\bar{A}$ has full row rank. Then the above system (3.1) has the unique solution*

$$\bar{d_x}^* := P_{\bar{A}}(-v), \;\; \textit{and} \;\; \bar{d_s}^* := (I - P_{\bar{A}})(-v),$$

where $P_{\bar{A}}$, which is defined in (2.1), is the orthogonal projection matrix onto $\mathcal{N}(\bar{A})$, the null space of $\bar{A}$.

Proof

First, by substituting $\bar{d_x}^* := P_{\bar{A}}(-v)$ and $\bar{d_s}^* := (I - P_{\bar{A}})(-v)$ into the system (3.1), we know that they provide a solution to the system (3.1). Second, we show the uniqueness of the solution. We know that $\bar{d_x}$ lies in the null space of $\bar{A}$ and $\bar{d_s}$ lies in the range of $\bar{A}^T$. Thus, by Lemma 2.1, we have $\bar{d_x}^T\bar{d_s}$=0. Suppose there exist another pair of solutions $\bar{d_x}'$ and $\bar{d_s}'$. Since $\bar{d_x}'$, $\bar{d_s}'$ also lie in the null space of $\bar{A}$ and the range of $\bar{A}^T$ respectively, we have that $\bar{d_x}' - d_x$, $\bar{d_s}' - d_s$ also lie in the null space of $\bar{A}$ and the range of $\bar{A}^T$ respectively. Thus $(\bar{d_x}' - \bar{d_x})^T(\bar{d_s}' - \bar{d_s}) = 0$. But, $\bar{d_x}' + \bar{d_s}' = \bar{d_x} + \bar{d_s} = -v$, so $0 = (\bar{d_x}' - \bar{d_x})^T(\bar{d_s}' - \bar{d_s}) = -\left\|\bar{d_s}' - \bar{d_s}\right\|$, which means $\bar{d_s}' = \bar{d_s}$. Similarly, $\bar{d_x}' = \bar{d_x}$. This proves the uniqueness of the solution. □

Now we can state the classical primal-dual affine-scaling method.

Algorithm 3.1
Input (A, x, s, ϵ)
while $x^T s > \epsilon$
$T := X^{1/2} S^{-1/2}$
$\bar{A} := AT$
solve the system (3.1) to get the unique solution $\bar{d_x}$ *and* $\bar{d_s}$
$d_x := T\bar{d_x}; \;\; d_s := T^{-1}\bar{d_s}$
find an $\alpha \geq 0$ *such that*
$x(\alpha) := x + \alpha d_x > 0$
$s(\alpha) := s + \alpha d_s > 0$
Let $x := x(\alpha); \;\; s := s(\alpha)$
repeat

This algorithm can be made to run in polynomial time (see Monteiro, Adler and Resende [22]).

3.2 The Zeroth-Order Algorithms

We may wonder about generalizing the above diagonal positive definite scaling matrix $X^{1/2}S^{-1/2}$ to any invertible scaling matrix T but keep the symmetric property. By symmetric, we mean that the primal variable x and the dual slack variable s are interchangeable. Symmetric algorithm will balance the primal and dual problem. A v-space approach (see Kojima, Megiddo, Noma, and Yoshise[14], and Tunçel [32]) is known to achieve this property. In such an approach, the scaling matrix T must satisfy the condition

$$Ts = v = \left(T^T\right)^{-1} x \, . \tag{3.2}$$

A direct consequence of the above identity (3.2) is that $(T^T T)s = x$. Since T is not singular, the linear transformation $(T^T T)$, mapping s to x, is a symmetric positive definite matrix. For every such transformation $(T^T T)$, there are many T exist. But if we force T to be also symmetric positive definite, then T is unique, which is the unique symmetric positive definite square root of the matrix $(T^T T)$. For simplification, we limit our discussion on the case that T is symmetric positive definite. Thus the linear transformation $(T^T T)$ can be written as T^2. Since the T and T^2 are uniquely determined by each other, we find it convenient to define T by explicitly describing T^2.

We use Σ^n_{++} to denote the set of symmetric positive definite matrices. For every pair of (x, s) which is strictly feasible solution for primal and dual respectively, we define

$$\mathcal{T}_0(x, s) := \{T \in \Sigma^n_{++} : T^2 s = x\}$$

and we state the zeroth-order algorithm(from [33]) as below:

Algorithm 3.2
Input (A, x, s, ϵ)
while $x^T s > \epsilon$
Find a $T \in \mathcal{T}_0(x, s)$
$\bar{A} := AT$
solve the system (3.1) to get the unique solution $\bar{d_x}$ *and* $\bar{d_s}$
$d_x := T\bar{d_x};\ \ d_s := T^{-1}\bar{d_s}$
find an $\alpha > 0$ *such that*
$x(\alpha) := x + \alpha d_x > 0$
$s(\alpha) := s + \alpha d_s > 0$
Let $x := x(\alpha);\ s := s(\alpha)$
repeat

We use the notation in the algorithm, namely $x(\alpha)$ for $(x + \alpha d_x)$ and $s(\alpha)$ for $(s + \alpha d_s)$. For the general zeroth-order algorithms, we have the following nice properties.

Theorem 3.2 *(Kojima and Tunçel [19]) Let* x, s *be strictly feasible solutions to the primal and the dual problems respectively. For every* $T \in \mathcal{T}_0(x, s)$*, the underlying search direction satisfies the following properties:*

1. $x(\alpha)^T s(\alpha) = (1 - \alpha)x^T s.$
2. *All primal-dual affine-scaling algorithms are strictly monotone in both primal and dual objectives, unless all the primal or all the dual solutions are optimal.*

3.3 Another Form of the Zeroth-Order Algorithms

We give another form of the zeroth-order algorithms at this section. By this form, we can easily see that the scaling matrix T is closely related to some ellipsoids inscribed in the feasible region.

The direction $\bar{d_x}$ given by the zeroth-order algorithms (Algorithm 3.2) can be directly calculated out by Lemma 3.1. The solution is $\bar{d_x} = -P_{\bar{A}}v$. It is also the optimal solution of the following problem:

$$\begin{aligned} \min v^T \bar{d_x} \ , & \\ \bar{A}\bar{d_x} &= 0 \ , \\ \|\bar{d_x}\|_2^2 &\leq \delta \ . \end{aligned} \tag{3.3}$$

Here, δ is some positive constant. The solution to the above problem is $\frac{-\delta P_{\bar{A}} v}{\|P_{\bar{A}} v\|}$.

The following problem directly comes from the above problem (3.3) by rewriting the scaling matrix T out explicitly.

$$\begin{aligned} \min s^T d_x \ , & \\ A d_x &= 0 \ , \\ {d_x}^T T^{-2} d_x &\leq \delta \ . \end{aligned} \tag{3.4}$$

Since $Ad_x = 0$, we have $s^T d_x = y^T A d_x + s^T d_x = c^T d_x$. Therefore, the above problem is equivalent to

$$\begin{aligned} \min c^T d_x , & \\ A d_x &= 0 \ , \\ {d_x}^T T^{-2} d_x &\leq \delta \ . \end{aligned} \tag{3.5}$$

From the above problem, we can see that the direction d_x is actually the optimal solution to a trust-region sub-problem using the steepest-descent direction. Roughly speaking, we want the trust-region specified by (${d_x}^T T^2 d_x \leq \delta$) to be a good approximation to the original feasible region.

Suppose we have a strictly feasible solution $(\hat{x}, \hat{s})$ at the current iteration. If we let δ be the maximum value such that $(\hat{x} + d_x)$ is in the primal feasible region, then the set $\{x \in \mathbb{R}^n : (x - \hat{x})^T T^{-2} (x - \hat{x}) \leq \delta, Ax = A\hat{x}\}$ defines an ellipsoid contained in the feasible region of (P). We showed that the optimal solution d_x in problem (3.5) is the same direction as the one defined in the zeroth-order algorithms (Algorithm 3.2). So, if we find a constant δ at each iteration and solve the problem (3.5) to get the optimal solution d_x, and update $\hat{x}$ to $\hat{x} + d_x$, then this algorithm, which defines a step length at each iteration while zeroth-order algorithms do not specify the step length, is actually a special case of the zeroth-order algorithms (Algorithm 3.2).

The zeroth-order algorithms are a generalization of the classical primal-dual affine scaling algorithm by generalize the scaling matrix T^2 from diagonal positive matrix to any positive definite one. By the above discussion, we can see that this generalization, in terms of geometric property, allows many more choices for the ellipsoid defined by the set $\{x \in \mathbb{R}^n : (x-\hat{x})^T T^{-2}(x-\hat{x}) \le \delta, Ax = A\hat{x}\}$.

The zeroth-order algorithms may not converge to an optimal solution in some cases. In the following section, we present an illustrative example showing the reason.

3.4 An Illustrative Example

Based on the last section's discussion, we give a concrete example in this section. We show why the algorithm may not converge in some cases.

Our example is based on the data:

$$\begin{aligned} A &:= \begin{pmatrix} -4 & 1 & -11 & 11 \\ -16 & 3 & 12 & 0 \end{pmatrix}, && (3.6)\\ x^0 &:= (1,1,1,1)^T, && (3.7)\\ s^0 &:= (1,1,1,1)^T. && (3.8) \end{aligned}$$

Moreover, the T^2 is defined as the following iterative formula:

$$T^2 := T_-^2 + \frac{xx^T}{x^T s} - \frac{T_-^2 s s^T T_-^2}{s^T T_-^2 s^T}. \tag{3.9}$$

This formula comes from [33]. Note that $T^2 s = x$, and T^2 is a symmetric positive definite matrix if T_-^2 is a symmetric positive definite matrix. We may let T_-^2 be the scaling matrix in the previous iteration. If it is the first iteration, we let T_-^2 be I. Also the step length is defined in the previous section, which is determined by the largest inscribed ellipsoid based on T^2.

Using the classical primal-dual affine-scaling interior-point method, we can find that the optimal solution of this problem is:

$$\begin{aligned} x^* : &= (0.2098,\ 0,\ 0.1964,\ 0)^T, \\ s^* : &= (0,\ 1.2187,\ 0,\ 2.375)^T. && (3.10) \end{aligned}$$

While using the T^2 defined by (3.9), we get the solution:

$$\begin{aligned} \hat{x} &:= (0.2299,\ 0.0997,\ 0.1982,\ 0)^T, \\ \hat{s} &:= (0.8920,\ 1.0231,\ 0.9239,\ 1.1235)^T. \end{aligned}$$

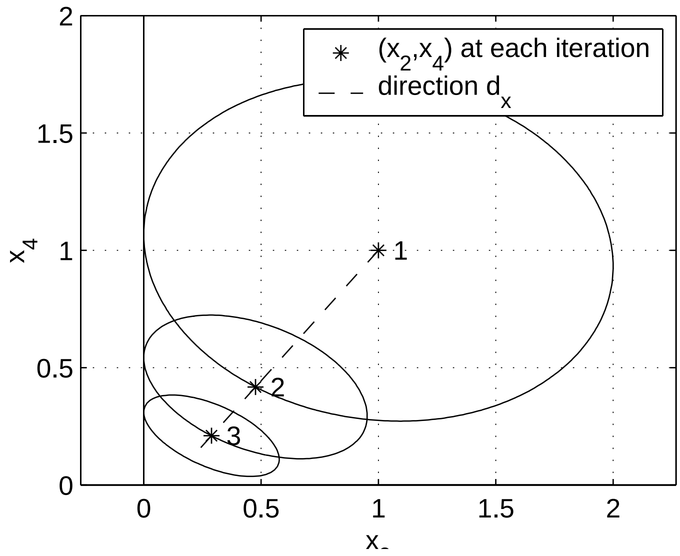

Figure 3.1: T^2 defined by (3.9)

Since $(x^*)^T\hat{s} > 0$ and $\hat{x}^T s^* > 0$, neither $\hat{x}$ nor $\hat{s}$ is optimal.

We projected the $\mathbb{R}^4_+$ onto the 2-dimensional space consisting of x_2 and x_4 and draw the figures. So the optimal solution should be at the origin in this (x_2, x_4) 2-dimensional space. Figure 3.1 and 3.2 show how the zeroth-order algorithms progress over iterations. We zoom in after iteration 3 on Figure 3.2 to make it easier to see. This algorithm uses the scaling matrix T defined in (3.9). Figure 3.3 and 3.4 also show how the algorithm works. But the difference is that we use the diagonal scaling matrix $T := X^{1/2}S^{-1/2}$ there. This scaling matrix turns the algorithm into the classical primal-dual affine-scaling algorithm.

First, we look at how the zeroth-order algorithms work. We can see that those ellipses do not change their shapes much after the first step. As the iterates tend to the boundary, the direction, which is determined by the objective function line and the ellipses, does not change much. This makes the step size smaller and smaller without much improvement in decreasing $x^T s$.

Let us have a look on Figure 3.3 and 3.4 to see how $T^2 := XS^{-1}$ works. We draw the picture on the same axes. We zoom in after the iteration 3 on Figure 3.4 to make it easier to see.

We can see that as x approaches to the axes x_2, the ellipses become skinnier and rotate so that the search direction, determined by the objective function and the ellipses, points closer to the origin, which is the optimal solution.

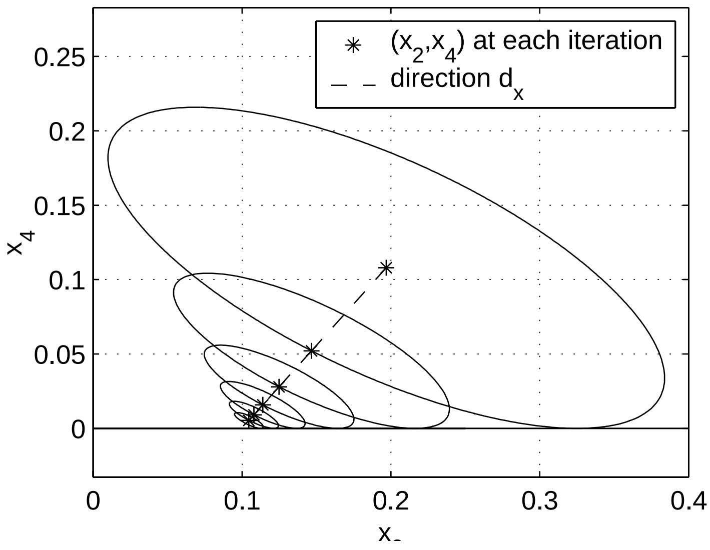

Figure 3.2: T^2 defined by (3.9), zoom in after iteration 3

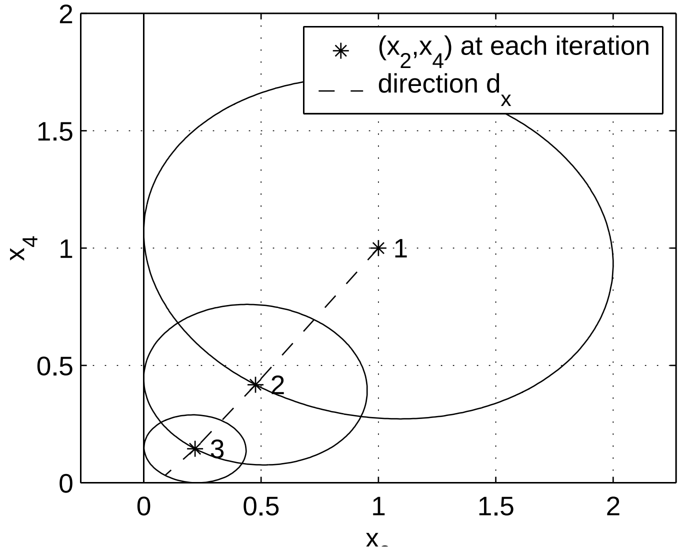

Figure 3.3: $T^2 := XS^{-1}$

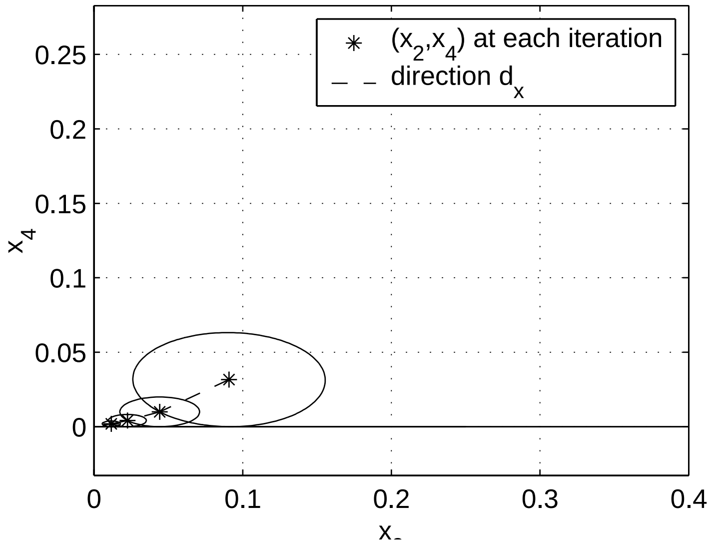

Figure 3.4: $T^2 := XS^{-1}$, zoom in after iteration 3

The key difference between the two algorithms is that when using the T^2 defined by (3.9), the ellipses do not change their shape and orientation much. Because the ellipses' shape and orientation are determined by the eigenvalues and eigenvectors of T^2, we conclude that T^2's eigenvalues do not change much through the iterations. The following sections of this chapter contain an algebraic proof of this fact.

3.5 Introduction to The Algebraic Proof

We consider the example based on the data given by (3.6), (3.7), and (3.8) in last section.

In every iteration, we define the step size α to be:

$$\alpha = 0.999\alpha_{max} , \tag{3.11}$$

where α_{max} is the maximum value of α such that $x + \alpha d_x \geq 0$, and $s + \alpha d_s \geq 0$. T^2 is defined by (3.9).

To be more clear, we state the algorithm here to specify how to choose T and how to determine the step length.

Algorithm 3.3

Input (A, x^0, s^0, ϵ), *where* A, x^0, s^0 *is defined in (3.6), (3.7), and (3.8)*

while $x^T s > \epsilon$

  if first iteration

    $T := I$

  else

    let T *defined by (3.9), where* T_-^2 *is the* T^2 *from the previous iteration*

  end if

  $\bar{A} := AT$

  solve the system (3.1) to get the unique solution $\bar{d}_x$ *and* $\bar{d}_s$

  $d_x := T\bar{d}_x; \;\; d_s := T^{-1}\bar{d}_s$

  use α *defined in (3.11)*

    $x(\alpha) := x + \alpha d_x > 0$

    $s(\alpha) := s + \alpha d_s > 0$

  Let $x := x(\alpha); \; s := s(\alpha)$

repeat

Theorem 3.3 *Algorithm 3.3 (a special case of the zeroth-order algorithms) does not converge to an optimal solution.*

We give a proof in the following 3 sections. Our discussion is based on the above algorithm. The proof is divided into two parts. In the first part, we give upper bounds on the quantities $\|T_+^2 - T^2\|$, $\|d_x^+ - d_x\|$, and $\|d_s^+ - d_s\|$. We use the variables with a superscript "+" to denote the next iteration. A special case is T_+^2, where we put "+" as the subscript instead of the superscript. In the second part, we stipulate two conditions, and prove that once the two conditions hold, they will keep holding for the remaining iterations. More importantly, we prove $\alpha^+ \leq \frac{2}{1000}\alpha$. Thus we get the result that Algorithm 3.3 does not converge.

3.6 Basic Theorems

First, we give some basic properties of eigenvalues and norms of a matrix in this section. We use Σ^n to denote the set of $n \times n$ symmetric real matrices. Let $A \in \Sigma^n$. We use $\lambda(A)$ to denote a vector whose entries are the eigenvalues of A. We use $\lambda_i(A)$ to denote the i^{th} largest eigenvalue of A. That is

$$\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_n(A) \; .$$

Theorem 3.4 (Courant-Fischer-Weyl Min-max Theorem) *Let $A \in \Sigma^n$, and L be a subspace in $\mathbb{R}^n$. Then*

$$\lambda_k(A) = \max_{dim(L)=k} \min_{x\in L\setminus\{0\}} \frac{x^TAx}{x^Tx} .$$

Setting $k = 1$ and $k = n$ in the above statement, we immediately obtain the following well-known fact.

Corollary 3.5 *Let $A \in \Sigma^n$. Then*

$$\begin{aligned} \lambda_1(A) &= \max_{x\in\mathbb{R}^n\setminus\{0\}} \frac{x^TAx}{x^Tx} , \\ \lambda_n(A) &= \min_{x\in\mathbb{R}^n\setminus\{0\}} \frac{x^TAx}{x^Tx} . \end{aligned}$$

By using Theorem 3.4, we have

$$\lambda_k(A+B) = \max_{dim(L)=k} \min_{x\in L\setminus\{0\}} \frac{x^T(A+B)x}{x^Tx} = \max_{dim(L)=k} \min_{x\in L\setminus\{0\}} \left(\frac{x^TAx}{x^Tx} + \frac{x^TBx}{x^Tx}\right) .$$

Combining the above fact with Corollary 3.5, we have the following corollary.

Corollary 3.6 *Let $A, B \in \Sigma^n$. Then*

$$\lambda_k(A) + \lambda_n(B) \le \lambda_k(A+B) \le \lambda_k(A) + \lambda_1(B) .$$

We define the norm of a matrix,

$$\|A\| := \max_{x\in\mathbb{R}^n\setminus\{0\}} \frac{\|Ax\|}{\|x\|},$$

where $A \in \mathbb{R}^{m\times n}$ and $x \in \mathbb{R}^n$. For all $A \in \Sigma^n$, we have

$$\|A\| = \max\{|\lambda_1(A)|\, , |\lambda_n(A)|\} . \tag{3.12}$$

Note that $\|A\|^2 = \max_{x\in\mathbb{R}^n\setminus\{0\}} \frac{\|Ax\|^2}{\|x\|^2} = \max_{x\in\mathbb{R}^n\setminus\{0\}} \frac{x^TA^TAx}{x^Tx} = \lambda_1(A^TA) = \lambda_1(A^2)$. Since A is a symmetric matrix, there exists an orthogonal matrix Q, such that $A = Q\text{Diag}(\lambda(A))Q^T$. Thus, $A^2 = Q\text{Diag}(\lambda(A))^2Q^T$. So, $\lambda_1(A^2) = \max\{(\lambda_1(A))^2, (\lambda_n(A))^2\}$. Hence, (3.12) follows.

Corollary 3.7 *Let A, $B \in \Sigma^n$. Then*

$$\lambda_k(A) \ge \lambda_k(B) - \|A - B\| .$$

Proof
Directly applying the identity (3.12) and Corollary 3.6, we obtain $\lambda_k(A) \geq \lambda_k(B) + \lambda_n(A-B) \geq \lambda_k(B) - \|A-B\|$. $\square$

3.7 Upper Bounds for the Quantities $\left\|T_+^2 - T^2\right\|$, $\|d_s^+ - d_s\|$, and $\|d_x^+ - d_x\|$.

Lemma 3.8 *Assume $0 \leq \alpha \leq 1/2$ and that T is defined by (3.9). Then*

$$\begin{aligned} &\|T_+^2 - T^2\| \\ \leq\;& \alpha\left(\frac{\|d_s\|\,\|x\|^3 + 2\|x\|^2\,\|s\|\,\|d_x\|}{x^T s\,\lambda_n(T^2)\,\|s^+\|^2} + \frac{4\,\|x\|\,\|d_x\| + \|d_x\|^2}{x^T s} + \frac{2\,\|x\|^2 + 2\,\|d_x\|\,\|x\| + \frac{1}{2}\,\|d_x\|^2}{\lambda_n(T^2)\,\|s^+\|^2}\right). \end{aligned}$$

Proof
By the update formula (3.9), it is clear that $\left\|T_+^2 - T^2\right\| = \left\|\frac{x^+(x^+)^T}{(x^+)^T s^+} - \frac{T^2 s^+ (s^+)^T T^2}{(s^+)^T T^2 s^+}\right\|$. We express $x^+ = x + \alpha d_x$, and $s^+ = s + \alpha d_s$, and we have

$$\frac{x^+(x^+)^T}{(x^+)^T s^+} = \frac{x x^T + \alpha(x d_x^T + d_x x^T) + \alpha^2 d_x d_x^T}{(1-\alpha) x^T s}.$$

Since $T^2 s^+ = T^2(s + \alpha d_s) = x + \alpha T^2 d_s$, by $\bar{d_x} + \bar{d_s} = -v$, we have $d_x + T^2 d_s = -x$, so $T^2 d_s = -x - d_x$. Thus, we have

$$T^2 s^+ = (1-\alpha)x - \alpha d_x. \tag{3.13}$$

It follows that

$$\frac{T^2 s^+ (s^+)^T T^2}{(s^+)^T T^2 s^+} = \frac{(1-\alpha)^2 x x^T - \alpha(1-\alpha)(d_x x^T + x d_x^T) + \alpha^2 d_x d_x^T}{(s^+)^T T^2 s^+}.$$

So,

$$\begin{aligned} &\left\|\frac{x^+(x^+)^T}{(x^+)^T s^+} - \frac{T^2 s^+ (s^+)^T T^2}{(s^+)^T T^2 s^+}\right\| \\ =\;& \Bigg\|\left(\frac{x x^T}{(1-\alpha) x^T s} - \frac{(1-\alpha)^2 x x^T}{(s^+)^T T^2 s^+}\right) + \frac{\alpha(x d_x^T + d_x x^T) + \alpha^2 d_x d_x^T}{(1-\alpha) x^T s} \\ &+ \frac{\alpha(1-\alpha)(d_x x^T + x d_x^T) - \alpha^2 d_x d_x^T}{(s^+)^T T^2 s^+}\Bigg\| \end{aligned}$$

$$\leq \left\| \frac{xx^T[(s^+)^T T^2 s^+ - (1-\alpha)^3 x^T s]}{(1-\alpha)x^T s\,(s^+)^T T^2 s^+} \right\| + \left\| \frac{\alpha(xd_x^T + d_x x^T) + \alpha^2 d_x d_x^T}{(1-\alpha)x^T s} \right\|$$
$$+ \left\| \frac{\alpha(1-\alpha)(d_x x^T + x d_x^T) - \alpha^2 d_x d_x^T}{(s^+)^T T^2 (s^+)} \right\|.$$

By using equation (3.13), $s^+ = s + \alpha d_x$, and $d_x^T d_s = 0$, we can simplify the first norm quantity of the right hand side to $\frac{xx^T[(1-\alpha)x^T s(2\alpha-\alpha^2)+\alpha(1-\alpha)d_s^T x - \alpha s^T d_x]}{(1-\alpha)x^T s\,(s^+)^T T^2 s^+}$. We take the norm inside to get a more relaxed bound:

$$\left\| \frac{x^+(x^+)^T}{(x^+)^T s^+} - \frac{T^2 s^+ (s^+)^T T^2}{(s^+)^T T^2 s^+} \right\|$$
$$\leq \alpha \Big(\frac{\|x\|^2 [(1-\alpha)\|d_s\|\,\|x\| + \|s\|\,\|d_x\|]}{(1-\alpha)x^T s\,(s^+)^T T^2 s^+} + \frac{2\|x\|\,\|d_x\| + \alpha\|d_x\|^2}{(1-\alpha)x^T s}$$
$$+ \frac{(2-\alpha)\|x\|^2 + 2(1-\alpha)\|d_x\|\,\|x\| + \alpha\|d_x\|^2}{(s^+)^T T^2 s^+} \Big).$$

By Corollary 3.5, we know that $(s^+)^T T^2 s^+ \geq \lambda_n(T^2)\,\|s^+\|^2$. We also have the assumption that $0 \leq \alpha \leq 1/2$. So we can simplify the above formula by using these two bounds and obtain the desired result. □

Corollary 3.9 *When A, x^0, s^0 is defined by (3.6), (3.7), and (3.8), we have*

$$\left\|T_+^2 - T^2\right\| \leq \alpha \left(\frac{512\,\|d_s\| + 1024\,\|d_x\|}{1.9 x^T s \lambda_n(T^2)} + \frac{32\,\|d_x\| + \|d_x\|^2}{x^T s} + \frac{128 + 16\,\|d_x\| + \frac{1}{2}\,\|d_x\|^2}{1.9\lambda_n(T^2)} \right).$$

Proof

For every $x \in \mathcal{F}(P)$ and $s \in \mathcal{F}(D)$, we have that $x - x^0$ is in the null space of A and $s - s^0$ is in the range of A^T. Thus $(x - x^0)^T(s - s^0) = 0$, which is equivalent to $(x - x^0)^T s = (x - s^0)^T s^0$. We obtain

$$\begin{aligned} x^T s^0 &= (x^0)^T s^0 + (x - x^0)^T s^0 \\ &= (x^0)^T s^0 + (x - x^0)^T s \\ &= (x^0)^T s^0 + x^T s - (x^0)^T s \\ &\leq (x^0)^T s^0 + x^T s \\ &\leq 2(x^0)^T s^0 \qquad \text{(by Theorem 3.2)} \\ &= 8\,. \end{aligned}$$

Similarly, $(x^0)^T s \leq 2(x^0)^T s^0 = 8$. Since $x \geq 0$ and $s \geq 0$, we have upper bounds of $\|x\| \leq 8$ and $\|s\| \leq 8$. A lower bound of $\|s\|$ can also be obtained. Since for any s feasible we have $(s - s^0)$ in the range of A^T, then $s = A^T y + s^0$ for some y. Thus, $\|s\| \geq \min_y \left\|A^T y + s^0\right\|$. $\min_y \left\|A^T y + s^0\right\|$ is a convex quadratic optimization problem. Based on the data of A and s^0, we get a numerical result $\min_y \left\|A^T y + s^0\right\| = 1.9893$. So we are safe to say $\|s\| \geq 1.9$. We substitute the bounds obtained here for $\|x\|$ and $\|s\|$ in the statement of Lemma 3.8 to get the result. □

We denote the upper bound given by the above corollary $Up\left(\left\|T_+^2 - T^2\right\|\right)$.

The following two lemmas will be used to prove an upper bound of $\|d_s^+ - d_s\|$.

Lemma 3.10 *Assume B and $C \in \mathbb{R}^{n\times n}$ nonsingular. Then*

$$\left\|B^{-1} - C^{-1}\right\| \leq \left\|B^{-1}\right\| \left\|C^{-1}\right\| \|B - C\| \,.$$

Proof

$\left\|B^{-1} - C^{-1}\right\| = \left\|B^{-1}(C - B)C^{-1}\right\| \leq \left\|B^{-1}\right\| \left\|C^{-1}\right\| \|B - C\|$. □

Lemma 3.11 *Given $T \in \Sigma_{++}^n$, suppose $A \in \mathbb{R}^{m\times n}$ has full row rank. Then*

$$\left\|(ATA^T)^{-1}\right\| \leq \frac{1}{\lambda_n(T)\lambda_n(AA^T)} \,.$$

Proof

For any $y \in \mathbb{R}^m \backslash \{0\}$, $y^T ATA^T y > 0$ and $y^T AA^T y > 0$, so (ATA^T) and $(AA^T) \in \Sigma_{++}^m$. Using Corollary 3.12, we have $\left\|(ATA^T)^{-1}\right\| = \lambda_1((ATA^T)^{-1}) = 1/\lambda_n(ATA^T)$. Moreover,

$$\begin{aligned}
\lambda_n(ATA^T) &= \min_{y\in\mathbb{R}^m\backslash\{0\}} \frac{y^T ATA^T y}{y^T y} \\
&= \min_{y\in\mathbb{R}^m\backslash\{0\}} \frac{y^T ATA^T y}{y^T AA^T y} \cdot \frac{y^T AA^T y}{y^T y} \\
&\geq \min_{y\in\mathbb{R}^m\backslash\{0\}} \frac{y^T ATA^T y}{y^T AA^T y} \min_{y\in\mathbb{R}^m\backslash\{0\}} \frac{y^T AA^T y}{y^T y} \\
&\geq \lambda_n(T)\lambda_n(AA^T) \,.
\end{aligned}$$

Therefore, the result follows. □

Lemma 3.12 *In the zeroth-order algorithm,*

$$\left\|d_s^+ - d_s\right\| \le \frac{\left\|T_+^2 - T^2\right\| \|A\|^3 \|b\|}{\lambda_n(T_+^2)\lambda_n(T^2)(\lambda_n(AA^T))^2}.$$

Proof
Directly solving the system (3.1), we obtain

$$d_s = -A(AT^2A^T)^{-1}b\,.$$

So,

$$\begin{aligned}
\left\|d_s^+ - d_s\right\| &= \left\|A[(AT_+^2A^T)^{-1} - (AT^2A^T)^{-1}]b\right\| \\
&\le \|A\|\,\|b\|\,\left\|(AT_+^2A^T)^{-1} - (AT^2A^T)^{-1}\right\| \\
&\le \|A\|\,\|b\|\,\left\|(AT_+^2A^T)^{-1}\right\| \left\|(AT^2A^T)^{-1}\right\| \left\|(AT_+^2A^T) - (AT^2A^T)\right\| \quad \text{(by Lemma 3.10)} \\
&\le \frac{\|A\|\,\|b\|\,\left\|AT_+^2A^T - AT^2A^T\right\|}{\lambda_n(T_+^2)\lambda_n(T^2)(\lambda_n(AA^T))^2} \qquad \text{(by Lemma 3.11)} \\
&\le \frac{\|A\|^3\,\|b\|\,\left\|T_+^2 - T^2\right\|}{\lambda_n(T_+^2)\lambda_n(T^2)(\lambda_n(AA^T))^2}\,.
\end{aligned}$$

□

Using the above lemma, and the upper bound on $\left\|T_+^2 - T^2\right\|$, we define an upper bound on $\|d_s^+ - d_s\|$ as:

$$Up\left(\left\|d_s^+ - d_s\right\|\right) := \frac{Up\left(\left\|T_+^2 - T^2\right\|\right)\|A\|^3\|b\|}{\lambda_n(T^2)\lambda_n(T_+^2)(\lambda_n(AA^T))^2}.$$

Lemma 3.13 *In the zeroth-order algorithm,*

$$\left\|d_x^+ - d_x\right\| \le \alpha\|d_x\| + \left\|T_+^2 - T^2\right\| \left\|d_s^+\right\| + \left\|T^2\right\| \left\|d_s^+ - d_s\right\|.$$

Proof
Since $\bar{d_x} + \bar{d_s} = -v$, we have $d_x = -x - T^2d_s$. So,

$$\begin{aligned}
\left\|d_x^+ - d_x\right\| &= \left\|(x^+ + T_+^2d_s^+) - (x + T^2d_s)\right\| \\
&= \left\|\alpha d_x + (T_+^2 - T^2)d_s^+ + T^2(d_s^+ - d_s)\right\| \\
&\le \|\alpha d_x\| + \left\|(T_+^2 - T^2)d_s^+\right\| + \left\|T^2(d_s^+ - d_s)\right\| \\
&\le \alpha\|d_x\| + \left\|T_+^2 - T^2\right\| \left\|d_s^+\right\| + \left\|T^2\right\| \left\|d_s^+ - d_s\right\|.
\end{aligned}$$

□

Using the above lemma, and both upper bounds on $\left\|T_+^2 - T^2\right\|$ and $\|d_s^+ - d_s\|$, we obtain an upper bound on $\|d_x^+ - d_x\|$ defined as:

$$Up(\left\|d_x^+ - d_x\right\|) := \alpha \left\|d_x\right\| + Up\left(\left\|T_+^2 - T^2\right\|\right) \left\|d_s^+\right\| + \left\|T^2\right\| Up\left(\left\|d_s^+ - d_s\right\|\right).$$

3.8 Progression of The Algorithm

We obtained the upper bounds for the quantities $\left\|T_+^2 - T^2\right\|$, $\|d_s^+ - d_s\|$, and $\|d_x^+ - d_x\|$ in the previous section. Now, we give two conditions, and prove that once these two conditions hold at the current iteration, they will hold throughout all the remaining iterations, and most importantly, α decreases very fast before the iterates approach an optimal solution.

Condition 1: There exists a unique index i^*, such that $0 < \alpha_{max} = \frac{x_{i^*}}{-d_{x_{i^*}}} \leq 1/2$ and $d_{x_{i^*}} < 0$, and

$$\begin{cases} \frac{x_i}{-d_{x_i}} > 2\alpha_{max} & \text{if } d_{x_i} < 0, \ i \neq i^* \\ \frac{s_i}{-d_{s_i}} > 2\alpha_{max} & \text{if } d_{s_i} < 0. \end{cases} \tag{3.14}$$

Condition 2:

$$Up\left(\left\|T_+^2 - T^2\right\|\right) \leq \frac{1}{2}\lambda_n(T^2), \tag{3.15}$$

$$Up\left(\left\|d_s^+ - d_s\right\|\right) \leq \frac{1}{2}\left|d_{s_i}\right| \quad \text{for all } i \ , \tag{3.16}$$

$$Up\left(\left\|d_x^+ - d_x\right\|\right) \leq \frac{1}{2}\left|d_{x_i}\right| \quad \text{for all } i \ . \tag{3.17}$$

Theorem 3.14 *Let A, x^0, s^0 be defined by (3.6),(3.7), and (3.8). Then for this problem, if Condition 1 and 2 hold at the current iteration, then they will also hold at the next iteration; moreover, ${\alpha_{max}}^+ \leq \frac{2}{1000}\alpha_{max}$.*

Proof

Suppose Condition 1 and 2 hold at the current iteration.

First, we consider the changes in x and s. Based on Condition 1 and that $\alpha = 0.999\alpha_{max}$, we have

$$\begin{aligned} x_{i^*}^+ &= x_{i^*} + \alpha d_{x_{i^*}} \\ &= x_{i^*} + 0.999\alpha_{max} d_{x_{i^*}} \end{aligned}$$

$$
\begin{aligned}
&= x_{i^*} + 0.999 \frac{x_{i^*}}{-d_{x_{i^*}}} d_{x_{i^*}} \\
&= \frac{1}{1000} x_{i^*} \,.
\end{aligned}
$$

For every $i \neq i^*$, if $d_{x_i} \geq 0$, then clearly $x_i^+ = x_i + \alpha d_{x_i} \geq x_i > \frac{1}{2} x_i$; if $d_{x_i} < 0$, then using Condition 1, we have:

$$
\begin{aligned}
x_i^+ &= x_i + \alpha d_{x_i} \\
&> x_i + \alpha_{max} d_{x_i} \\
&> x_i - \frac{1}{2} x_i = \frac{1}{2} x_i .
\end{aligned}
$$

In summary, it is

$$
\begin{cases} x_{i^*}^+ = \frac{1}{1000} x_{i^*} & \\ x_i^+ \geq \frac{1}{2} x_i & \text{for every } i \neq i^*. \end{cases} \tag{3.18}
$$

By applying the same calculation on s, we obtain

$$
s_i^+ \geq \frac{1}{2} s_i, \text{ for all } i. \tag{3.19}
$$

Second, we consider the changes in d_x and d_s. Using Condition 2, we have

$$
\left\| d_x^+ - d_x \right\| \leq Up \left(\left\| d_x^+ - d_x \right\| \right) \leq \frac{1}{2} \left| d_{x_i} \right| .
$$

Also, clearly

$$
d_{x_i} - \left\| d_x^+ - d_x \right\| \leq d_{x_i}^+ \leq d_{x_i} + \left\| d_x^+ - d_x \right\| .
$$

Thus,

$$
\begin{cases} \frac{1}{2} d_{x_i} \leq d_{x_i}^+ \leq \frac{3}{2} d_{x_i} & \text{if } d_{x_i} \geq 0, \\ \frac{3}{2} d_{x_i} \leq d_{x_i}^+ \leq \frac{1}{2} d_{x_i} & \text{if } d_{x_i} < 0. \end{cases} \tag{3.20}
$$

Applying the same calculation on d_s , we have:

$$
\begin{cases} \frac{1}{2} d_{s_i} \leq d_{s_i}^+ \leq \frac{3}{2} d_{s_i} & \text{if } d_{s_i} \geq 0 \,, \\ \frac{3}{2} d_{s_i} \leq d_{s_i}^+ \leq \frac{1}{2} d_{s_i} & \text{if } d_{s_i} < 0 \,. \end{cases} \tag{3.21}
$$

These inequalities mean that throughout the rest of the iterations d_{x_i}, d_{s_i} do not change their signs.

Now, we consider the change in α_{max}. First, we focus on the index i^*. Using (3.18) and (3.20)

$$
\frac{{x_{i^*}}^+}{-d_{x_{i^*}}^+} \leq \frac{\frac{1}{1000} x_{i^*}}{-\frac{1}{2} d_{x_{i*}}} = \frac{2}{1000} \alpha_{max} .
$$

Those indexes i such that $d_{x_i} > 0$ have no relation with the determination of α_{max}. We only consider those indexes i such that $d_{x_i} < 0$. For these indexes, using the relation (3.18), (3.20), and condition 1, we have

$$\frac{x_i^+}{-d_{x_i}^+} \geq \frac{\frac{1}{2}x_i}{-\frac{3}{2}d_{x_i}} \geq \frac{2}{3}\alpha_{max} \ \text{, for every } i \neq i^*,\ d_{x_i} < 0;$$

Similarly, for the s-space, we obtain

$$\frac{s_i^+}{-d_{s_i}^+} \geq \frac{2}{3}\alpha_{max} \ \text{, for every } i,\ d_{s_i} < 0.$$

So, by the definition of α_{max}, the only index that determines α_{max}^+ is the index i^*:

$$\alpha_{max}{}^+ = \frac{x_{i^*}{}^+}{-d_{x_{i^*}}^+} \leq \frac{2}{1000}\alpha_{max}. \tag{3.22}$$

Thus, Condition 1 holds at the next iteration too.

Now, we prove that Condition 2 holds. First we show that the first inequality (3.15) of Condition 2 holds. We use "++" as superscript or subscript to denote those variables in the iteration after the next iteration.

Using Corollary 3.7 and Condition 2, we can describe a lower bound on $\lambda_n(T^2)$:

$$\lambda_n(T_+^2) \geq \lambda_n(T^2) - \left\|T_+^2 - T^2\right\| \geq \lambda_n(T^2) - Up\left(\left\|T_+^2 - T^2\right\|\right) \geq \frac{1}{2}\lambda_n(T^2). \tag{3.23}$$

Now, using the definition of $Up\left(\left\|T_+^2 - T^2\right\|\right)$, we have

$$\begin{aligned} Up\left(\left\|T_+^2 - T^2\right\|\right) &= \alpha\left(\frac{512\left\|d_s\right\| + 1024\left\|d_x\right\|}{1.9x^Ts\lambda_n(T^2)} + \frac{32\left\|d_x\right\| + \left\|d_x\right\|^2}{x^Ts} + \frac{128 + 16\left\|d_x\right\| + \frac{1}{2}\left\|d_x\right\|^2}{1.9\lambda_n(T^2)}\right); \\ Up\left(\left\|T_{++}^2 - T_+^2\right\|\right) &= \alpha^+\left(\frac{512\left\|d_s^+\right\| + 1024\left\|d_x^+\right\|}{1.9(x^+)^Ts^+\lambda_n(T_+^2)} + \frac{32\left\|d_x^+\right\| + \left\|d_x^+\right\|^2}{(x^+)^Ts^+} + \frac{128 + 16\left\|d_x^+\right\| + \frac{1}{2}\left\|d_x^+\right\|^2}{1.9\lambda_n(T_+^2)}\right). \end{aligned}$$

We compare

$$\begin{aligned} \alpha^+ &\leq \frac{2}{1000}\alpha, \text{ by (3.22), and } \alpha = 0.999\alpha_{max}\ ; \\ \left\|d_s^+\right\| &\leq \frac{3}{2}\left\|d_s\right\|, \text{ by (3.21);} \\ \left\|d_x^+\right\| &\leq \frac{3}{2}\left\|d_x\right\|, \text{ by (3.20);} \\ (x^+)^Ts^+ &= (1-\alpha)x^Ts \geq \frac{1}{2}x^Ts, \text{ by the assumption } \alpha_{max} < 1/2 \text{ in Condition 1;} \\ \lambda_n(T_+^2) &\geq \frac{1}{2}\lambda_n(T^2), \text{ by (3.23).} \end{aligned}$$

So, we have

$$Up\left(\left\|T_{++}^2 - T_+^2\right\|\right) \leq \frac{\frac{2}{1000} \times (\frac{3}{2})^2}{\frac{1}{2} \times \frac{1}{2}} Up\left(\left\|T_+^2 - T^2\right\|\right) = \frac{18}{1000} Up\left(\left\|T_+^2 - T^2\right\|\right). \tag{3.24}$$

Now, using (3.24) and Condition 2, we can provide an upper bound on the change in T_+^2 in terms of the smallest eigenvalue of T^2.

$$Up\left(\left\|T_{++}^2 - T_+^2\right\|\right) \leq \frac{18}{1000} Up\left(\left\|T_+^2 - T^2\right\|\right) \leq \frac{18}{1000}\frac{1}{2}\lambda_n(T^2) = \frac{9}{1000}\lambda_n(T^2). \tag{3.25}$$

Using the relation (3.23) and the above result, we can prove that the first inequality (3.15) of the Condition 2 holds at the next iteration:

$$Up\left(\left\|T_{++}^2\right\| - T_+^2\right) \leq \frac{9}{1000}\lambda_n(T^2) \leq \frac{1}{2}\lambda_n(T_+^2)$$

Now, we prove that the second inequality (3.16) of Condition 2 holds. Using Corollary 3.7, the relations (3.23) and (3.25), we obtain

$$\lambda_n(T_{++}^2) \geq \lambda_n(T_+^2) - \left\|T_{++}^2 - T_+^2\right\| \geq \frac{1}{2}\lambda_n(T^2) - \frac{9}{1000}\lambda_n(T^2) = \frac{491}{1000}\lambda_n(T^2).$$

By the above fact and (3.24), we have

$$\begin{aligned}
Up\left(\left\|d_s^{++} - d_s^+\right\|\right) &= \frac{Up(\left\|T_{++}^2 - T_+^2\right\|)\,\|A\|^3\,\|b\|}{\lambda_n(T_+^2)\lambda_n(T_{++}^2)(\lambda_n(AA^T))^2} \\
&= \frac{Up(\left\|T_{++}^2 - T_+^2\right\|)\lambda_n(T^2)}{Up(\left\|T_+^2 - T^2\right\|)\lambda_n(T_{++}^2)} Up\left(\left\|d_s^+ - d_s\right\|\right) \\
&\leq \frac{18}{1000} \times \frac{1000}{491} Up\left(\left\|d_s^+ - d_s\right\|\right) && (3.26)\\
&\leq \frac{18}{491}\frac{1}{2}\left|d_{s_i}\right| \leq \frac{1}{2}\left|d_{s_i}^+\right|, \quad \text{for all } i\,. && (3.27)
\end{aligned}$$

We used the inequality (3.21) at the last step. So, the second inequality (3.16) of Condition 2 holds.

Now, considering $\|d_x^{++} - d_x^+\|$, we know

$$\begin{aligned}
Up\left(\left\|d_x^+ - d_x\right\|\right) &= \alpha\,\|d_x\| + Up\left(\left\|T_+^2 - T^2\right\|\right)\left\|d_s^+\right\| + \left\|T^2\right\| Up\left(\left\|d_s^+ - d_s\right\|\right); \\
Up\left(\left\|d_x^{++} - d_x^+\right\|\right) &= \alpha^+\left\|d_x^+\right\| + Up\left(\left\|T_{++}^2 - T_+^2\right\|\right)\left\|d_s^{++}\right\| + \left\|T_+^2\right\| Up\left(\left\|d_s^{++} - d_s{}^+\right\|\right).
\end{aligned}$$

We compare again:

$$\alpha^+ \leq \frac{2}{1000}\alpha, \text{ by Condition 1;}$$

$$
\begin{aligned}
\left\|d_x^+\right\| &\leq \frac{3}{2}\left\|d_x\right\|, \text{ by } (3.20); \\
Up\left(\left\|T_{++}^2 - T_+^2\right\|\right) &\leq \frac{18}{1000} Up\left(\left\|T_+^2 - T^2\right\|\right), \text{ by } (3.24); \\
\left\|d_s^{++}\right\| &\leq \left\|d_s^+\right\| + \left\|d_s^{++} - d_s^+\right\| \leq \left\|d_s^+\right\| + \frac{1}{2}\min_i \left|d_{s_i}^+\right| \leq \frac{3}{2}\left\|d_s^+\right\|, \text{ by } (3.27); \\
\left\|T_+^2\right\| &\leq \left\|T^2\right\| + \left\|T_+^2 - T^2\right\| \leq \left\|T^2\right\| + \frac{1}{2}\lambda_n(T^2) \leq \frac{3}{2}\left\|T^2\right\|, \text{ by Condition 2}; \\
Up\left(\left\|d_s^{++} - d_s^+\right\|\right) &\leq \frac{18}{491} Up\left(\left\|d_s^+ - d_s\right\|\right), \text{ by } (3.26).
\end{aligned}
$$

By using above listed relations, Condition 2, and (3.20), we obtain

$$
Up\left(\left\|d_x^{++} - d_x^+\right\|\right) \leq \frac{18}{491}\frac{3}{2} Up\left(\left\|{d_x}^+ - d_x\right\|\right) \leq \frac{27}{491}\frac{1}{2}\left|d_{x_i}\right| \leq \frac{1}{2}\left|{d_{x_i}}^+\right|.
$$

Therefore, we proved that all of the conditions hold at the next iteration, and that $\alpha_{max}^+ \leq (2/1000)\alpha_{max}$.

□

We have found an iteration that satisfies Conditions 1 and 2 in Matlab. In fact, it is pretty easy to find, because α always decreases quickly, while others, like x, s, d_x, d_s, and T^2 do not change significantly when α goes to zero.

The following list presents the results from Matlab 5.3 on a common PC with 500MHz CPU. We can see that Condition 1 and 2 are both satisfied at iteration 4, and the index i^* is 4.

Iterations	1	2	3	4
α	0.8765	0.0076	7.6909×10^{-6}	7.6909×10^{-9}
$Up(\|T_+^2 - T^2\|)$	336.5851	28.3690	0.0289	2.8971×10^{-5}
$Up(\|d_s^+ - d_s\|)$	425.4212	317.0502	0.3253	3.2534×10^{-4}
$Up(\|d_x^+ - d_x\|)$	491.1124	352.3395	0.3621	3.6216×10^{-4}
$\lambda_n(T^2)$	1	0.1140	0.1131	0.1131
d_s	$\begin{pmatrix} -0.1222 \\ 0.0261 \\ -0.0862 \\ 0.1397 \end{pmatrix}$	$\begin{pmatrix} -0.1126 \\ -0.0240 \\ -0.0793 \\ 0.1287 \end{pmatrix}$	$\begin{pmatrix} -0.1124 \\ -0.0240 \\ -0.0792 \\ 0.1284 \end{pmatrix}$	$\begin{pmatrix} -0.1124 \\ -0.0240 \\ -0.0792 \\ 0.1284 \end{pmatrix}$
d_x	$\begin{pmatrix} -0.8778 \\ -1.0261 \\ -0.9138 \\ -1.1397 \end{pmatrix}$	$\begin{pmatrix} -0.1009 \\ -0.1179 \\ -0.1050 \\ -0.1310 \end{pmatrix}$	$\begin{pmatrix} -0.1000 \\ -0.1169 \\ -0.1041 \\ -0.1299 \end{pmatrix}$	$\begin{pmatrix} -0.1000 \\ -0.1169 \\ -0.1041 \\ -0.1299 \end{pmatrix}$
s	$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0.8929 \\ 1.0229 \\ 0.9245 \\ 1.1225 \end{pmatrix}$	$\begin{pmatrix} 0.8920 \\ 1.0231 \\ 0.9239 \\ 1.1235 \end{pmatrix}$	$\begin{pmatrix} 0.8920 \\ 1.0231 \\ 0.9239 \\ 1.1235 \end{pmatrix}$
x	$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0.2306 \\ 0.1006 \\ 0.1990 \\ 0.0010 \end{pmatrix}$	$\begin{pmatrix} 0.2299 \\ 0.0997 \\ 0.1982 \\ 1 \times 10^{-6} \end{pmatrix}$	$\begin{pmatrix} 0.2299 \\ 0.0997 \\ 0.1982 \\ 1 \times 10^{-9} \end{pmatrix}$

This numerical result gives a proof that at iteration 4, Condition 1 and 2 holds. By Theorem 3.14, we know that these two conditions will still hold through the rest iterations and the step length will decrease more than 99.8% at each iteration. Using the optimal value x^* and s^* given in (3.10) , we find $x^T s^* = 0.3686$, and $(x^*)^T s = 0.1215$, which are both far away from 0. So the x and s at iteration 4 are both not optimal solutions. At iteration 4, the decrease of the duality gap is $\alpha x^T s = 7.6909 \times 10^{-9} \times x^T s = 3.7697 \times 10^{-9}$. Since α_{max} decreases more than 99.8% at each iteration, the total decrease of the duality gap through the rest of iteration will be no more than a very small value, which will not match current duality gap $x^T s^* = 0.3686$, and $(x^*)^T s = 0.1215$. we conclude that the algorithm will not converge to an optimal solution. This proves Theorem 3.3.

Chapter 4

Scaling Matrices for the Second-Order Algorithm

4.1 Introduction to the First-Order Algorithms

We showed in the last chapter that the zeroth-order algorithms do not converge to an optimal solution in some cases. We may want to add more limitations on the choices of scaling matrices, or add the centering direction to improve the algorithm. The centering direction will use the first order information of the barrier functions. By using the first order information of the barrier functions, we have a globally-convergent algorithm. We do not give any new result in this section. They are a restatement of Tunçel [33]'s general first-order algorithms in linear programming form.

Recall $\tilde{x} := S^{-1}e$ and $\tilde{s} := X^{-1}e$. Then $\tilde{x}$ and $\tilde{s}$ are the gradients of the barrier functions for primal and dual problems respectively. We also denote $\tilde{\mu} := \tilde{x}^T\tilde{s}/n$. Now, we can define the set of the scaling matrices T in the first-order algorithms. For every pair $(x, s) \in \mathcal{F}_+(P) \oplus \mathcal{F}_+(D)$, We define

$$\mathcal{T}_1(x, s) := \{T \in \Sigma^n_{++} : T^2 s = x, T^2\tilde{s} = \tilde{x}\}.$$

Theorem 4.1 *For every pair $(x, s) \in \mathcal{F}_+(P) \oplus \mathcal{F}_+(D)$, if $x = t\tilde{x}$ for some $t > 0$, then*

$$\mathcal{T}_1(x, s) = \mathcal{T}_0(x, s);$$

otherwise, define

$$T_H^2 := H + a_1 xx^T + g_1 Hss^T H + \tilde{a}_1\tilde{x}\tilde{x}^T + \tilde{g}_1 H\tilde{s}\tilde{s}^T H + a_2(x\tilde{x}^T + \tilde{x}x^T) + g_2(Hs\tilde{s}^T H + H\tilde{s}s^T H),$$

where

$$a_1 := \frac{\tilde{\mu}}{n(\mu\tilde{\mu}-1)}\,,\ \tilde{a}_1 := \frac{\mu}{n(\mu\tilde{\mu}-1)}\,,\ a_2 := \frac{1}{n(\mu\tilde{\mu}-1)},$$

$$g_1 := -\frac{\tilde{s}^T H\tilde{s}}{(s^T Hs)(\tilde{s}^T H\tilde{s}) - (\tilde{s}^T Hs)^2}\,,$$

$$\tilde{g}_1 := -\frac{s^T Hs}{(s^T Hs)(\tilde{s}^T H\tilde{s}) - (\tilde{s}^T Hs)^2}\,,\ \ \tilde{g}_2 := -\frac{\tilde{s}^T Hs}{(s^T Hs)(\tilde{s}^T H\tilde{s}) - (\tilde{s}^T Hs)^2}\,.$$

Then

$$T_H \in \mathcal{T}_1(x,s),\ \textit{for every } H \in \Sigma^n_{++}\,.$$

Now, we define $w := T\tilde{s} = T^{-1}\tilde{x}$. The search direction is described as by the following system.

$$\begin{aligned} \bar{A}\bar{d}_x &= 0, \\ \bar{A}^T d_y + \bar{d}_s &= 0, \\ \bar{d}_x + \bar{d}_s &= -v + \gamma\mu w, \end{aligned} \tag{4.1}$$

where $\bar{A} := AT$, $\bar{d}_x := T^{-1}d_x$, $\bar{d}_s := Td_s$, and $\gamma \in [0,1]$, a centering parameter.

Theorem 4.2 *Let $\gamma \in [0,1]$ and $T \in \mathcal{T}_1(x,s)$. Then*

$$x(\alpha)^T s(\alpha) = [1 - \alpha(1-\gamma)]x^T s.$$

Now, we can state the first-order algorithm.

Algorithm 4.1
Input: $(A,\ x^0,\ s^0,\ \gamma,\ \rho,\ \epsilon)$
while $x^T s > \epsilon$
  choose $T \in \mathcal{T}_1(x,s)$
  compute $v := Ts$, $w := T\tilde{s}$, *and* $\bar{A} := AT$
  solve the system (4.1) to get the unique solution $\bar{d}_x$ *and* $\bar{d}_s$
  $d_x := T\bar{d}_x;\ \ d_s := T^{-1}\bar{d}_s$
  find $\hat{\alpha}$ *such that* $\phi_\rho(x(\hat{\alpha}), s(\hat{\alpha}) = \min_{\alpha>0}\phi_\rho(x(\alpha), s(\alpha))$ *(for* $\phi_\rho(x,s)$*, see Definition 2.11)*
  Let $x := x(\hat{\alpha});\ \ s := s(\hat{\alpha})$
repeat

Theorem 4.3 *All first-order algorithms with fixed $\gamma \in [0,1)$ and $\rho > 0$ are globally convergent.*

Setting $\gamma = 0$ results a zeroth-order algorithm with more restriction on the scaling matrices T as well as allowing line search on a potential function to determine the step size α. This restriction make the zeroth-order algorithms globally convergent.

4.2 Introduction to the Second-Order Algorithms

First-order algorithms have global convergence result. But we do not know if the first-order algorithms have polynomial iteration bound. Second-order algorithms further restrict the choices of scaling matrices in first-order algorithms by using the Hessians of the barrier functions. All second-order algorithms have polynomial-iteration bound. A lemma in the next chapter (Lemma 5.5) gives an explanation that the new added restriction to the first-order algorithms actually gives a lower bound on the step size α.

We define

$$\delta_F(x,s) := \frac{1}{\mu}[n(\mu\tilde{\mu} - 1) + 1] \geq \frac{1}{\mu} > 0$$

(We used Lemma 2.10). The Hessians of the barrier functions are X^{-2} and S^{-2} respectively. Now, we add a new constraint to the set $\mathcal{T}_1(x,s)$ using the Hessians of the barrier functions , and by this way, we get the scaling matrices set in second-order algorithms. We define

$$\mathcal{T}_2(\xi; x, s) := \{T \in \Sigma^n_{++} : T^2 s = x,\, T^2 X^{-1} e = S^{-1} e,\, \frac{1}{\xi \delta_F(x,s)} S^{-2} \preceq T^2 \preceq \xi \delta_F(x,s) X^2\} \quad (4.2)$$

where ξ is a positive constant.

We also define $v := Ts$, $w := T\tilde{s}$,

$$\tilde{u} := -\Big(\frac{n + \sqrt{n}}{n\mu}\Big) v + w, \qquad \text{and } u := \frac{\tilde{u}}{\|\tilde{u}\|_2}.$$

Note that

$$\|\tilde{u}\|_2^2 = \delta_F(x,s) > 0;$$

therefore, u is well defined. Then we solve the system of linear equations

$$\begin{aligned} \bar{A}\bar{d}_x &= 0, \\ \bar{A}^T d_y + \bar{d}_s &= 0, \\ \bar{d}_x + \bar{d}_s &= u. \end{aligned} \quad (4.3)$$

Now, we can state the second-order algorithm.

Algorithm 4.2
Input (A, x, s, ϵ, ξ)
while $x^T s > \epsilon$
choose $T \in \mathcal{T}_2(\xi; x, s)$
compute $v := Ts$, $w := T\tilde{s}$, $\tilde{u}$, *and* u
$\bar{A} := AT$
solve the system (4.3) to get the unique solution $\bar{d}_x$ *and* $\bar{d}_s$
$d_x := T\bar{d}_x$; $d_s := T^{-1}\bar{d}_s$
find a $\hat{\alpha} \geq 0$ *to minimize the potential function* $\phi_{\sqrt{n}}(x(\hat{\alpha}), s(\hat{\alpha}))$ *(see Definition 2.11)*
Let $x := x + \hat{\alpha} d_x$; $s := s + \hat{\alpha} d_s$
repeat

The above algorithm is a restatement of the more general second-order algorithm in Tunçel [33]. We apply the algorithm specifically to the linear programming, while the original second-order algorithms can be applied to any convex optimization programming in conic form.

Tunçel [33] proved that for any $T \in \mathcal{T}_2(\xi; x, s)$ with ξ a constant, all second-order algorithms have polynomial-iteration bounds.

It follows from Theorem 6.1 of [33] that the choice

$$\hat{T} := X^{1/2} S^{-1/2},$$

lies in the set $\mathcal{T}_2(\frac{4}{3}; x, s)$.

Now the problem is that given the constant $\xi \geq \frac{4}{3}$ and the variables x and s, can we find all of the $T \in \mathcal{T}_2(\xi; x, s)$? The next section answers this question.

4.3 Scaling Matrices in the Second-Order Algorithms

In this section, we give some theorems to show how to construct a matrix T in $\mathcal{T}_2(\xi; x, s)$. Although we can not find a way to construct all the matrices in $\mathcal{T}_2(\xi; x, s)$, we give a theorem to bound the $\mathcal{T}_2(\xi; x, s)$.

We write $T^2 = XS^{-1} + E$, where $E \in \Sigma^n$. Then $T \in \mathcal{T}_2(\xi; x, s)$ if and only if $E \in \mathcal{E}(\xi; x, s)$, where

$$\mathcal{E}(\xi; x, s) := \{E \in \Sigma^n : Es = EX^{-1}e = 0, \underline{E} \preceq E \preceq \bar{E}\},$$

and

$$\begin{aligned} \bar{E} &:= \xi\delta_F(x,s)X^2 - XS^{-1}, \\ \underline{E} &:= \frac{1}{\xi\delta_F(x,s)}S^{-2} - XS^{-1}. \end{aligned}$$

Since $\hat{T} = X^{1/2}S^{-1/2}$ lies in the set $\mathcal{T}_2(\frac{4}{3}; x, s)$, and $X,\ S$ are both positive definite, with $\xi > 4/3$, $\bar{E}$ is always positive definite, and $\underline{E}$ is always negative definite.

Also, we can define the sets

$$\begin{aligned} \mathcal{E}^+(\xi; x, s) &:= \left\{E \in \Sigma^n : Es = EX^{-1}e = 0,\ 0 \preceq E \preceq \bar{E}\right\}, \\ \mathcal{E}^-(\xi; x, s) &:= \left\{E \in \Sigma^n : Es = EX^{-1}e = 0,\ \underline{E} \preceq E \preceq 0\right\}. \end{aligned}$$

The following lemma gives a method to construct all matrix in $\mathcal{E}^+(\xi; x, s)$ or $\mathcal{E}^-(\xi; x, s)$.

Lemma 4.4 *Assume $\bar{E}$ is positive definite. Then $E \in \mathcal{E}^+(\xi; x, s)$ if and only if*

$$0 \le \lambda(\bar{E}^{-1/2}E\bar{E}^{-1/2}) \le e, \quad E \in \Sigma^n$$

and $(\bar{E}^{-1/2}E\bar{E}^{-1/2})(\bar{E}^{1/2}s) = (\bar{E}^{-1/2}E\bar{E}^{-1/2})(\bar{E}^{1/2}X^{-1}e) = 0$.

Proof
Notice that

$$\begin{aligned} & 0 \preceq E \preceq \bar{E} \\ \iff\ & 0 \preceq \bar{E}^{-1/2}E\bar{E}^{-1/2} \preceq I \\ \iff\ & 0 \le \lambda[\bar{E}^{-1/2}E\bar{E}^{-1/2}] \le e\,. \end{aligned}$$

Also, because $\bar{E}^{-1/2}$ is a positive diagonal matrix,

$$\bar{E}^{-1/2}E\bar{E}^{-1/2}h = 0 \iff E(\bar{E}^{-1/2}h) = 0 \tag{4.4}$$

Let $\bar{E}^{-1/2}h = s$, we have $h = \bar{E}^{1/2}s$. Similarly, we have $h = \bar{E}^{1/2}X^{-1}e$. So $Es = EX^{-1}e = 0$ is equivalent to $(\bar{E}^{-1/2}E\bar{E}^{-1/2})(\bar{E}^{1/2}s) = (\bar{E}^{-1/2}E\bar{E}^{-1/2})(\bar{E}^{1/2}X^{-1}e) = 0$. □

We also have a similar theorem on the set $\mathcal{E}^-(\xi; x, s)$.

Lemma 4.5 *Assume $\underline{E}$ is negative definite. Then $E \in \mathcal{E}^-(\xi; x, s)$ if and only if*

$$0 \geq \lambda((-\underline{E})^{-1/2} E (-\underline{E})^{-1/2}) \geq -e \tag{4.5}$$

and $[(-\underline{E})^{-1/2} E (-\underline{E})^{-1/2}][(-\underline{E})^{1/2} s] = [(-\underline{E})^{-1/2} E (-\underline{E})^{-1/2}][(-\underline{E})^{1/2} X^{-1} e] = 0.$

Proof
Note that $\underline{E} \preceq E \preceq 0$ is equivalent to $0 \preceq -E \preceq -\underline{E}$. Applying Lemma 4.4, we have the desired result. □

Theorem 4.6 *Assuming $\bar{E}$ is positive definite and $\underline{E}$ is negative definite, a sufficient condition for $E \in \mathcal{E}(\xi; x, s)$ is:*

$$\frac{1}{\xi \delta_F(x,s) \max_i\{x_i s_i\}} e \leq \lambda(\bar{E}^{-1/2} E \bar{E}^{-1/2}) \leq e$$

and $(\bar{E}^{-1/2} E \bar{E}^{-1/2})(\bar{E}^{1/2} s) = (\bar{E}^{-1/2} E \bar{E}^{-1/2})(\bar{E}^{1/2} X^{-1} e) = 0.$

A necessary condition for $E \in \mathcal{E}(\xi; x, s)$ is:

$$\frac{1}{\xi \delta_F(x,s) \min_i\{x_i s_i\}} e \leq \lambda(\bar{E}^{-1/2} E \bar{E}^{-1/2}) \leq e$$

and $(\bar{E}^{-1/2} E \bar{E}^{-1/2})(\bar{E}^{1/2} s) = (\bar{E}^{-1/2} E \bar{E}^{-1/2})(\bar{E}^{1/2} X^{-1} e) = 0.$

Proof
First, we see that $\bar{E}^{-1/2} E \bar{E}^{-1/2} h = 0 \iff E[\bar{E}^{-1/2} h] = 0$, so, $Es = 0$ and $EX^{-1}e = 0$ is equivalent to $(\bar{E}^{-1/2} E \bar{E}^{-1/2})(\bar{E}^{1/2} s) = (\bar{E}^{-1/2} E \bar{E}^{-1/2})(\bar{E}^{1/2} X^{-1} e) = 0$.

Second,

$$\underline{E} \preceq E \preceq \bar{E} \iff \bar{E}^{-1/2} \underline{E} \bar{E}^{-1/2} \preceq \bar{E}^{-1/2} E \bar{E}^{-1/2} \preceq I\,.$$

So a sufficient condition for $\underline{E} \leq E \leq \bar{E}$ is that

$$\lambda_1(\bar{E}^{-1/2} \underline{E} \bar{E}^{-1/2}) e \leq \lambda \bar{E}^{-1/2} E \bar{E}^{-1/2} \leq e, \tag{4.6}$$

and a necessary condition for $\underline{E} \leq E \leq \bar{E}$ is that

$$\lambda_n(\bar{E}^{-1/2} \underline{E} \bar{E}^{-1/2}) e \leq \lambda \bar{E}^{-1/2} E \bar{E}^{-1/2} \leq e. \tag{4.7}$$

Using the fact that both $\underline{E}$ and $\bar{E}$ are diagonal matrices we obtain

$$\bar{E}^{-1/2} \underline{E} \bar{E}^{-1/2} = \bar{E}^{-1} \underline{E} = (\xi \delta_F(x,s) X^2 - XS^{-1})^{-1} (\frac{1}{\xi \delta_F(x,s)} S^{-2} - XS^{-1})\,.$$

So,

$$[\bar{E}^{-1/2}\underline{E}\bar{E}^{-1/2}]_{ii} = \frac{x_i s_i^{-1}(\frac{1}{\xi\delta_F(x,s)}x_i^{-1}s_i^{-1} - 1))}{x_i s_i^{-1}(\xi\delta_F(x,s)x_i s_i - 1)} = \frac{-1}{\xi\delta_F(x,s)x_i s_i}.$$

Thus,

$$\lambda_1[\bar{E}^{-1/2}\underline{E}\bar{E}^{-1/2}] = \frac{-1}{\xi\delta_F(x,s)\max_i\{x_i,s_i\}}, \tag{4.8}$$

$$\lambda_n[\bar{E}^{-1/2}\underline{E}\bar{E}^{-1/2}] = \frac{-1}{\xi\delta_F(x,s)\min_i\{x_i,s_i\}}. \tag{4.9}$$

Combining (4.6), (4.7), (4.8) and (4.9), we have the desired conclusion. □

Define $\mathcal{E}^+(\xi;x,s) + \mathcal{E}^-(\xi;x,s) := \{E^+ + E^- : E^+ \in \mathcal{E}^+(\xi;x,s), E^- \in \mathcal{E}^-(\xi;x,s)\}$.

Theorem 4.7

$$\mathcal{E}(\xi;x,s) \supseteq \mathcal{E}^+(\xi;x,s) + \mathcal{E}^-(\xi;x,s);$$

and

$$\mathcal{E}(\xi;x,s) \subseteq \frac{\max_i\{x_i s_i\}}{\min_i\{x_i s_i\}}\mathcal{E}^+(\xi;x,s) + \mathcal{E}^-(\xi;x,s),$$

or

$$\mathcal{E}(\xi;x,s) \subseteq \mathcal{E}^+(\xi;x,s) + \frac{\max_i\{x_i s_i\}}{\min_i\{x_i s_i\}}\mathcal{E}^-(\xi;x,s).$$

Especially, when $s = tX^{-1}e$ for some $t > 0$, $\mathcal{E}(\xi;x,s) = \mathcal{E}^+(\xi;x,s) + \mathcal{E}^-(\xi;x,s)$.

Proof

Picking any matrix $E^+ \in \mathcal{E}^+(\xi;x,s)$, and any matrix $E^- \in \mathcal{E}^-(\xi;x,s)$, we have

$$\underline{E} \preceq E^- \preceq E^+ + E^- \preceq E^+ \preceq \bar{E}.$$

Also, $(E^+ + E^-)s = E^+s + E^-s = 0$, and $(E^+ + E^-)X^{-1}e = 0$. Thus $(E^+ + E^-) \in \mathcal{E}(\xi;x,s)$ which means $\mathcal{E}(\xi;x,s) \supseteq \mathcal{E}^+(\xi;x,s) + \mathcal{E}^-(\xi;x,s)$.

Conversely, for any $E \in \mathcal{E}(\xi;x,s)$, $\bar{E}^{-1/2}E\bar{E}^{-1/2}$ is still a symmetric matrix. We can write $\bar{E}^{-1/2}E\bar{E}^{-1/2} = U\Lambda U^T$, where U is an orthogonal matrix and Λ is a diagonal matrix. Let Λ^+, and Λ^- be two matrices to denote the non-negative part and the non-positive part of matrix Λ respectively, that is $\Lambda^+_{ij} = \Lambda_{ij}$ if $\Lambda_{ij} \geq 0$, $\Lambda^+_{ij} = 0$ if $\Lambda_{ij} < 0$; $\Lambda^-_{ij} = \Lambda_{ij}$ if $\Lambda_{ij} \leq 0$, $\Lambda^-_{ij} = 0$ if $\Lambda_{ij} > 0$. Thus, $\Lambda = \Lambda^+ + \Lambda^-$. We denote $E^+ := \bar{E}^{1/2}U\Lambda^+U^T\bar{E}^{1/2}$, and $E^- := \bar{E}^{1/2}U\Lambda^-U^T\bar{E}^{1/2}$, then $E = E^+ + E^-$. Because $\lambda(\bar{E}^{-1/2}E\bar{E}^{-1/2}) = \lambda(\Lambda^+)$, by Lemma 4.4, we know that $E^+ \in \mathcal{E}^+(\xi;x,s)$.

Because $E \in \mathcal{E}(\xi; x, s)$, using the necessary condition in theorem 4.6, we have $\Lambda \succeq \frac{-1}{\xi\delta_F(x,s)\min_i\{x_i s_i\}} I$. So, we have

$$\Lambda^- \succeq \frac{-1}{\xi\delta_F(x,s)\min_i\{x_i s_i\}} I.$$

This is equivalent to

$$\frac{\min_i\{x_i s_i\}}{\max_i\{x_i s_i\}}\Lambda^- \succeq \frac{-1}{\xi\delta_F(x,s)\max_i\{x_i s_i\}} I,$$

which, by Theorem 4.6, is a sufficient condition for $\frac{\min_i\{x_i s_i\}}{\max_i\{x_i s_i\}} E^- \in \mathcal{E}^-(\xi; x, s)$. Thus, we have $\mathcal{E}(\xi; x, s) \subseteq \mathcal{E}^+(\xi; x, s) + \frac{\max_i\{x_i s_i\}}{\min_i\{x_i s_i\}}\mathcal{E}^-(\xi; x, s)$.

Similarly, we prove that $\mathcal{E}(\xi; x, s) \subseteq \frac{\max_i\{x_i s_i\}}{\min_i\{x_i s_i\}}\mathcal{E}^+(\xi; x, s) + \mathcal{E}^-(\xi; x, s)$.

When $s = tX^{-1}e$ for some $t > 0$, $\max_i\{x_i s_i\} = \min_i\{x_i s_i\}$. Thus, $\mathcal{E}(\xi; x, s) = \mathcal{E}^+(\xi; x, s) + \mathcal{E}^-(\xi; x, s)$. □

The matrices in the set $\mathcal{E}^+(\xi; x, s)$ and $\mathcal{E}^-(\xi; x, s)$ are very easy to construct. Given x and s at the current iteration, we can first calculate the matrix $\bar{E}$. Since $\bar{E}^{-1/2} E \bar{E}^{-1/2}$ can be written as $U\Lambda U^T$, we are free to pick the diagonal entries of Λ, the only constraint is that there must be at least one zero on the diagonal entries and $I \geq \Lambda \geq 0$. As for the orthogonal matrix U, the only limitation is that $\bar{E}^{1/2}s$ and $\bar{E}^{1/2}X^{-1}e$ must lie in the zero eigenspace. After choosing the appropriate Λ and U, we have $E^+ := \bar{E}^{-1/2} U\Lambda U^T \bar{E}^{1/2}$ which is a matrix in the set $\mathcal{E}^+(\xi; x, s)$. Similarly, we can construct a matrix E^- in the set $\mathcal{E}^-(\xi; x, s)$, so that $(E^+ + E^- + XS^{-1})$ is in $\mathcal{T}_2(\xi; x, s)$.

Chapter 5

On the Proximity Measures for the Central Path

We showed that $\mu\tilde{\mu} \geq 1$ in Chapter 2(see Lemma 2.10). Furthermore, the inequality holds as equality if and only if $s = tX^{-1}e$ for some $t > 0$, in words, if and only if x and s lie on the central path (assuming x and s are strictly feasible in the corresponding problems). So, $(\mu\tilde{\mu} - 1)$ is a centrality measure. Also, in the first-order algorithms, $(\mu\tilde{\mu} - 1)$ appears at the iterative formula in Theorem 4.1. In the second-order algorithms, we also have $(\mu\tilde{\mu} - 1)$ appearing in the definition of $\xi\delta_F(x, s)$. Studying $(\mu\tilde{\mu} - 1)$ further may help us improve our understanding of the first- and the second-order algorithms. One thing we need to know is how the neighbourhood defined by this centrality measure relates to the old neighbourhoods, such as those we defined in Chapter 2. Another thing we want to know is how $\mu\tilde{\mu}$ changes in the next iteration in terms of the current iteration's information.

Recall that we gave some of the neighbourhoods of the central path in Chapter 2. One of the neighbourhoods is

$$\mathcal{N}_\infty^-(\beta) := \{(x, s) \in \mathcal{F}_+ : \|Xs - \mu e\|_\infty^- \leq \beta\mu\}.$$

Here, for $v \in \mathbb{R}^n$, $\|v\|_\infty^- := -\min\{0, \ \min_j\{v_j\}\}$. We also define a neighbourhood based on $\mu\tilde{\mu}$:

$$\mathcal{N}_{\mu\tilde{\mu}}(\beta) := \{(x, s) \in \mathcal{F}_+ : \mu\tilde{\mu} \leq \frac{1}{1-\beta}\} \qquad \text{for } \beta \in [0, 1).$$

We have the following relation between the new neighbourhood $\mathcal{N}_{\mu\tilde{\mu}}(\beta)$ and the old neighbourhood $\mathcal{N}_\infty^-(\beta)$.

Theorem 5.1 *Assume the dimension $n \geq 2$. For a given β such that $0 \leq \beta < 1$, we have*

$$\mathcal{N}_\infty^-(\beta) \subseteq \mathcal{N}_{\mu\tilde{\mu}}(\beta) \subseteq \mathcal{N}_\infty^-\Big(\sqrt{[1+\beta(\frac{n}{2}-1)]^2 - (1-\beta)} - \beta(\frac{n}{2}-1)\Big).$$

Proof

By the definition of norm $\|\cdot\|_\infty^-$, we have $(x,s) \in \mathcal{N}_\infty^-(\beta)$ if and only if $(\min_i\{x_i s_i\}/\mu) \geq 1-\beta$. Therefore, $\mu\tilde{\mu} \leq \frac{\mu}{\min_i\{x_i s_i\}} \leq \frac{1}{1-\beta}$. This means $(x,s) \in \mathcal{N}_{\mu\tilde{\mu}}(\beta)$. So we have $\mathcal{N}_\infty^-(\beta) \subseteq \mathcal{N}_{\mu\tilde{\mu}}(\beta)$.

For any $(x,s) \in \mathcal{N}_{\mu\tilde{\mu}}(\beta)$, we have $\mu\tilde{\mu} \leq \frac{1}{1-\beta}$. Using Lemma 2.9, the Arithmetic-Harmonic Mean Inequality, we have

$$\begin{aligned}
\frac{1}{1-\beta} &\geq \mu\tilde{\mu} \\
&= \mu\frac{1}{n}\sum_{i=1}^{n}\frac{1}{x_i s_i} \\
&\geq \mu\frac{1}{n}\Big(\frac{1}{\min_i\{x_i s_i\}} + \frac{(n-1)^2}{n\mu - \min_i\{x_i s_i\}}\Big) \quad \text{(by Arithmetic-Harmonic Mean Inequality)} \\
&= \frac{1}{n}\Big(\frac{\mu}{\min_i\{x_i s_i\}} + \frac{(n-1)^2}{n - \min_i\{x_i s_i\}/\mu}\Big).
\end{aligned}$$

We consider $\min_i\{x_i s_i\}/\mu$ as a variable in $[0,1]$, then the above inequality is a quadratic inequality. We directly solve the quadratic inequality, and get the answer:

$$\frac{\min_i\{x_i s_i\}}{\mu} \geq [1+\beta(\frac{n}{2}-1)] - \sqrt{[1+\beta(\frac{n}{2}-1)]^2 - (1-\beta)}\,.$$

So, we have the desired result. □

We write $x(\alpha)$ for $(x+\alpha d_x)$, and $s(\alpha)$ for $(s+\alpha d_s)$. The corresponding shadow variables at the next iteration are $\tilde{x}(\alpha) := \text{Diag}(s(\alpha))^{-1}e$ and $\tilde{s}(\alpha) := \text{Diag}(x(\alpha))^{-1}e$. Then we define $\mu(\alpha) := x(\alpha)s(\alpha)/n$, and $\tilde{\mu}(\alpha) := \tilde{x}(\alpha)\tilde{s}(\alpha)/n$.

The following lemma gives an upper bound to $\tilde{\mu}(\alpha)$ in terms of the step length α and the current iteration's information.

Lemma 5.2 *Assume α is a step length such that $\tilde{\mu}(\alpha)$ is meaningful, that is $x(\alpha), s(\alpha)$ are both strictly feasible. Then*

$$\tilde{\mu}(\alpha) \leq \tilde{\mu}(0) + \alpha\tilde{\mu}'(0) + \frac{2\alpha^2}{(1-\alpha/\alpha_{max})^4}\tilde{\mu}''(0),$$

where $\alpha_{max} := \max\{\alpha \geq 0 : x+\alpha d_x \geq 0, s+\alpha d_s \geq 0\}$.

Proof

By the definition of α_{max}, we have

$$x_i(\alpha) = x_i + \alpha d_{x_i} \geq x_i - \alpha \frac{x_i}{\alpha_{max}} = (1 - \frac{\alpha}{\alpha_{max}})x_i > 0.$$

Similarly,

$$s_i(\alpha) = s_i + \alpha d_{s_i} \geq (1 - \frac{\alpha}{\alpha_{max}})s_i > 0.$$

As for $\tilde{\mu}''(\alpha)$, we have

$$n\tilde{\mu}''(\alpha) = \sum_{i=1}^{n} \frac{1}{x_i(\alpha)s_i(\alpha)} \left[\left(\frac{d_{x_i}}{x_i(\alpha)} + \frac{d_{s_i}}{s_i(\alpha)} \right)^2 + \left(\frac{d_{x_i}}{x_i(\alpha)} \right)^2 + \left(\frac{d_{s_i}}{s_i(\alpha)} \right)^2 \right]. \tag{5.1}$$

By replacing each $x_i(\alpha)$ with $(1 - \frac{\alpha}{\alpha_{max}})x_i$, $s_i(\alpha)$ with $(1 - \frac{\alpha}{\alpha_{max}})s_i$, we obtain a new function:

$$f_i(\alpha) \quad := \quad \frac{1}{(1 - \frac{\alpha}{\alpha_{max}})^4} \frac{1}{x_i s_i} \left[\left(\frac{d_{x_i}}{x_i} + \frac{d_{s_i}}{s_i} \right)^2 + \left(\frac{d_{x_i}}{x_i} \right)^2 + \left(\frac{d_{s_i}}{s_i} \right)^2 \right], \tag{5.2}$$

Note that

$$\sum_{i=1}^{n} f_i(\alpha) \quad = \quad \frac{1}{(1 - \frac{\alpha}{\alpha_{max}})^4} n\mu''(0). \tag{5.3}$$

If d_{x_i} and d_{s_i} have the same signs, which means $d_{x_i}d_{s_i} \geq 0$, then

$$\frac{1}{x_i(\alpha)s_i(\alpha)} \left[\left(\frac{d_{x_i}}{x_i(\alpha)} + \frac{d_{s_i}}{s_i(\alpha)} \right)^2 + \left(\frac{d_{x_i}}{x_i(\alpha)} \right)^2 + \left(\frac{d_{s_i}}{s_i(\alpha)} \right)^2 \right] \leq f_i(\alpha). \tag{5.4}$$

Otherwise, if d_{x_i} and d_{s_i} have different signs, which means $d_{x_i}d_{s_i} < 0$, then

$$\left(\frac{d_{x_i}}{x_i(\alpha)} + \frac{d_{s_i}}{s_i(\alpha)} \right)^2 = \left(\frac{d_{x_i}}{x_i(\alpha)} \right)^2 + \left(\frac{d_{s_i}}{s_i(\alpha)} \right)^2 + \frac{2d_{x_i}d_{s_i}}{x_i(\alpha)s_i(\alpha)} < \left(\frac{d_{x_i}}{x_i(\alpha)} \right)^2 + \left(\frac{d_{s_i}}{s_i(\alpha)} \right)^2. \tag{5.5}$$

So,

$$\begin{aligned} & \frac{1}{x_i(\alpha)s_i(\alpha)} \left[\left(\frac{d_{x_i}}{x_i(\alpha)} + \frac{d_{s_i}}{s_i(\alpha)} \right)^2 + \left(\frac{d_{x_i}}{x_i(\alpha)} \right)^2 + \left(\frac{d_{s_i}}{s_i(\alpha)} \right)^2 \right] \\ \leq \; & \frac{2}{x_i(\alpha)s_i(\alpha)} \left[\left(\frac{d_{x_i}}{x_i(\alpha)} \right)^2 + \left(\frac{d_{s_i}}{s_i(\alpha)} \right)^2 \right] \\ \leq \; & 2f_i(\alpha). \end{aligned}$$

By (5.1), (5.3), (5.4), and (5.5), we conclude that:

$$\tilde{\mu}''(\alpha) \leq \frac{2}{n}\sum_{i=1}^{n} f_i(\alpha) = \frac{2\tilde{\mu}''(0)}{(1-\frac{\alpha}{\alpha_{max}})^4}.$$

By Taylor's theorem, we have:

$$\begin{aligned}
\tilde{\mu}(\alpha) &= \tilde{\mu}(0) + \alpha\tilde{\mu}'(0) + \frac{\alpha^2}{2}\tilde{\mu}''(\beta) \qquad \text{for some } \beta \in [0,\alpha] \\
&\leq \tilde{\mu}(0) + \alpha\tilde{\mu}'(0) + \frac{2\alpha^2}{(1-\frac{\beta}{\alpha_{max}})^4}\tilde{\mu}''(0) \\
&\leq \tilde{\mu}(0) + \alpha\tilde{\mu}'(0) + \frac{2\alpha^2}{(1-\frac{\alpha}{\alpha_{max}})^4}\tilde{\mu}''(0) \quad (\text{by } \beta \leq \alpha).
\end{aligned}$$

□

Note that $\mu(\alpha)$ can be expressed in terms of μ. More specifically (see [33]), in the zeroth-order algorithms.

$$\mu(\alpha) = (1-\alpha)\mu;$$

in the first-order algorithms:

$$\mu(\alpha) = [1-\alpha(1-\gamma)]\mu;$$

in the second-order algorithms:

$$\mu(\alpha) = \left[1 - \alpha\frac{1}{\sqrt{n\delta_F(x,s)\mu}}\right]\mu, \tag{5.6}$$

where $\delta_F(x,s) := \frac{1}{\mu}[n(\mu\tilde{\mu}-1)+1]$. So an upper bound of $\tilde{\mu}(\alpha)$ gives an upper bound of $\mu(\alpha)\tilde{\mu}(\alpha)$.

Also there is an interesting relation between the α_{max} and the second-order algorithm. The following well-known relation can be found in Todd's survey paper [30].

Lemma 5.3 $\frac{1}{\alpha_{max}} \leq \max\{\|d_x\|_x, \|d_x\|_s\}$

Proof
Define

$$\begin{aligned}
\sigma_x(d_x) &:= \min\{\sigma \geq 0, \quad \sigma x + d_x \geq 0\}; \\
\sigma_s(d_s) &:= \min\{\sigma \geq 0, \quad \sigma s + d_s \geq 0\}; \\
\sigma &:= \max\{\sigma_x(d_x), \quad \sigma_s(d_s)\}.
\end{aligned}$$

We have $\sigma = \frac{1}{\alpha_{max}}$. Also, we have

$$\begin{aligned} |d_x|_x &= \max\{\sigma_x(d_x), \quad \sigma_x(-d_x)\} = \|X^{-1}d_x\|_\infty, \\ \|d_x\|_x &= \|X^{-1}d_x\|_2 = (d_x^T X^{-2} d_x)^{\frac{1}{2}}. \end{aligned}$$

So, we have $\sigma_x(d_x) \leq |d_x|_x \leq \|d_x\|_x$. Similarly, $\sigma_s(d_s) \leq |d_s|_s \leq \|d_s\|_s$. That is $\frac{1}{\alpha_{max}} = \sigma \leq \max\{\|d_x\|_x, \|d_s\|_s\}$. □

The following lemma comes from Tunçel [33].

Lemma 5.4 *In Algorithm 4.2, the second-order algorithm,*

$$\|d_x\|_x^2 \leq \xi\delta_F(x,s) \quad \textit{and} \quad \|d_s\|_s^2 \leq \xi\delta_F(x,s).$$

Proof

By the definition of the norm $\|\cdot\|_x$, we have

$$\|d_x\|_x^2 = \bar{d_x}^T T X^{-2} T \bar{d_x} \leq \|\bar{d_x}\|_2^2 \|TX^{-2}T\|_2 \leq \|TX^{-2}T\|_2.$$

We have used the property that $\|\bar{d_x}\|_2 \leq 1$ in the second-order algorithm. Since $T \in \mathcal{T}_2(\xi; x, s)$, by the definition of $\mathcal{T}_2(\xi; x, s)$ (4.2), we know that $T^2 \preceq \xi\delta_F(x,s)X^2$, which is equivalent to $TX^{-2}T \preceq \xi\delta_F(x,s)I$. Hence, $\|TX^{-2}T\|_2 \leq \xi\delta_F(x,s)$. That is $\|d_x\|_x^2 \leq \xi\delta_F(x,s)$.

Similarly we have the desired result of $\|d_s\|_s^2 \leq \xi\delta_F(x,s)$. □

Directly using the Lemma 5.4 and Lemma 5.3, we conclude the following lemma.

Lemma 5.5 *In the second-order algorithm, we have*

$$\frac{1}{\alpha_{max}} \leq \sqrt{\xi\delta_F(x,s)}\,.$$

This lemma combined with Lemma 5.4 also give an explanation that the new added constraint $(\frac{1}{\xi\delta_F(x,s)}S^{-2} \preceq T^2 \preceq \xi\delta_F(x,s)X^2)$ in the second-order algorithms gives a lower bound to the step length α.

Theorem 5.6 *Suppose α is a step length such that $\tilde{\mu}(\alpha)$ is meaningful. Then in the second-order algorithm, we have*

$$\frac{\mu(\alpha)\tilde{\mu}(\alpha)}{\mu\tilde{\mu}} \leq \Big(1 - \frac{\alpha}{\sqrt{n\delta_F(x,s)\mu}}\Big)\Big(1 + 2\sqrt{\xi\delta_F(x,s)} + \frac{12\xi\delta_F(x,s)\alpha^2}{(1-\alpha/\alpha_{max})^4}\Big).$$

Proof

We use Lemma 5.2 to prove this theorem. In the second-order algorithms, we have

$$\begin{aligned}\frac{d_{x_i}}{x_i} &\leq \|d_x\|_x \qquad \text{(by the definition of norm } \|\cdot\|_x) \\ &\leq \sqrt{\xi \delta_F(x,s)} \qquad \text{(using Lemma 5.4)}.\end{aligned}$$

Similarly,

$$\frac{d_{si}}{s_i} \leq \sqrt{\xi \delta_F(x,s)}.$$

Therefore,

$$\begin{aligned}\tilde{\mu}'(0) &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{x_i s_i}\left(\frac{d_{x_i}}{x_i} + \frac{d_{s_i}}{s_i}\right) \\ &\leq \frac{1}{n}\sum_{i=1}^{n}\frac{1}{x_i s_i} 2\sqrt{\xi \delta_F(x,s)} \\ &= 2\sqrt{\xi \delta_F(x,s)}\tilde{\mu}.\end{aligned}$$

Similarly,

$$\begin{aligned}\tilde{\mu}''(0) &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{x_i s_i}\left[\left(\frac{d_{x_i}}{x_i} + \frac{d_{s_i}}{s_i}\right)^2 + \left(\frac{d_{x_i}}{x_i}\right)^2 + \left(\frac{d_{s_i}}{s_i}\right)^2\right] \\ &\leq 6\xi \delta_F(x,s)\tilde{\mu}.\end{aligned}$$

Now, combining Lemma 5.2 and the fact (5.6), we obtain the desired result. □

Bibliography

[1] E. R. Barnes. A variation on Karmarkar's algorithm for solving linear programming problems. *Mathematical Programming*, 36:174–182, 1986.

[2] G. Dantzig. Maximization of a linear function of variables subject to linear ineqaulities. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 339–347. John Wiley, New York, 1951.

[3] G. B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, Princeton, NJ, 1963, 1963.

[4] I. I. Dikin. Iterative solution of problems of linear and quadratic programming. *Doklady Akademii Nauk SSSR*, 174:747–748, 1967. Translated in: *Soviet Mathematics Doklady* 8:674–675, 1967.

[5] R. M. Freund. Polynomial–time algorithms for linear programming based only on primal scaling and projected gradients of a potential function. *Mathematical Programming*, 51:203–222, 1991.

[6] C. C. Gonzaga. An algorithm for solving linear programming problems in $O(n^3L)$ operations. In N. Megiddo, editor, *Progress in Mathematical Programming: Interior Point and Related Methods*, pages 1–28. Springer Verlag, New York, 1989.

[7] C. C. Gonzaga. Polynomial affine algorithms for linear programming. *Mathematical Programming*, 49:7–21, 1990.

[8] C. C. Gonzaga. Path following methods for linear programming. *SIAM Review*, 34(2):167–224, 1992.

[9] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz. An interior–point method for semidefinite programming. *SIAM Journal on Optimization*, 6:342–361, 1996.

[10] N. K. Karmarkar. A new polynomial–time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[11] L. G. Khachiyan. A polynomial algorithm for linear progrmming. *Soviet Math. Dokl.*, 20:191–194, 1979.

[12] L. G. Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Math. Phys.*, 20:53–72, 1980.

[13] V. Klee and G. Minty. How good is the simplex algorithm? In O. Sisha, editor, *Inequalities III*. Academic Press, New York, NY, 1972.

[14] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise. A unified approach to interior point algorithms for linear complementarity problems: A summary. *Operations Research Letters*, 10:247–254, 1991.

[15] M. Kojima, S. Mizuno, and A. Yoshise. A polynomial–time algorithm for a class of linear complementarity problems. *Mathematical Programming*, 44:1–26, 1989.

[16] M. Kojima, S. Mizuno, and A. Yoshise. A primal–dual interior point algorithm for linear programming. In N. Megiddo, editor, *Progress in Mathematical Programming: Interior Point and Related Methods*, pages 29–47. Springer Verlag, New York, 1989.

[17] M. Kojima, S. Mizuno, and A. Yoshise. An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems. *Mathematical Programming*, 50:331–342, 1991.

[18] M. Kojima, S. Shindoh, and S. Hara. Interior–point methods for the monotone semidefinite linear complementarity problem in symmetric matrices. *SIAM Journal on Optimization*, 7:86–125, 1997.

[19] M. Kojima and L. Tunçel. Monotonicity of primal-dual interior-point algorithms for semidefinite programming problems. *Optimization Methods and Software*, 10:275–296, 1998.

[20] L. McLinden. The analogue of Moreau's proximation theorem, with applications to the nonlinear complementarity problem. *Pacific Journal of Mathematics*, 88:101–161, 1980.

[21] R. D. C. Monteiro and I. Adler. Interior path following primal–dual algorithms, Part I: Linear programming. *Mathematical Programming*, 44:27–41, 1989.

[22] R. D. C. Monteiro, I. Adler, and M. G. C. Resende. A polynomial–time primal–dual affine scaling algorithm for linear and convex quadratic programming and its power series extension. *Mathematics of Operations Research*, 15:191–214, 1990.

[23] R. D. C. Monteiro and Y. Zhang. A unified analysis for a class of path–following primal–dual interior–point algorithms for semidefinite programming. *Mathematical Programming*, 81:281–299, 1998.

[24] Yu. E. Nesterov and A. S. Nemirovskii. *Interior–Point Polynomial Algorithms in Convex Programming : Theory and Algorithms*, volume 13 of *Studies in Applied Mathematics.* Society of Industrial and Applied Mathematics (SIAM) Publications, Philadelphia, PA 19101, USA, 1993.

[25] Yu. E. Nesterov and M. J. Todd. Self–scaled barriers and interior–point methods for convex programming. *Mathematics of Operations Research*, 22:1–42, 1997.

[26] Yu. E. Nesterov and M. J. Todd. Primal–dual interior–point methods for self–scaled cones. *SIAM Journal on Optimization*, 8:324–364, 1998.

[27] J. Renegar. A polynomial–time algorithm based on Newton's method for linear programming. *Mathematical Programming*, 40:59–93, 1988.

[28] N. Shor. Utilization of the operation of space dilatation in the minimization of convex functions. *Kibernetika*, 1:6–12, 1970. (In Russian). Translated in : *Cybernetics, 6,* 7-15.

[29] K. Tanabe. Complementarity–enforced centered Newton method for mathematical programming. In K. Tone, editor, *New Methods for Linear Programming*, pages 118–144, The Institute of Statistical Mathematics, 4–6–7 Minami Azabu, Minatoku, Tokyo 106, Japan, 1987.

[30] M. J. Todd. Potential–reduction methods in mathematical programming. *Mathematical Programming*, 76:3–45, 1997.

[31] M. J. Todd and Y. Ye. A centered projective algorithm for linear programming. *Mathematics of Operations Research*, 15:508–529, 1990.

[32] L. Tunçel. Primal-dual symmetry and scale invariance of interior-point algorithms for convex optimization. *Mathematics of Operations Research*, 23:708–718, 1998.

[33] L. Tunçel. Generalization of primal-dual interior-point methods to convex optimization problems in conic form. *Foundations of Computational Mathematics*, 1:229–254, 2001.

[34] P. M. Vaidya. An algorithm for linear programming which requires $O((m+n)n^2 + (m+n)^{1.5}nL)$ arithmetic operations. *Mathematical Programming*, 47:175–201, 1990. Condensed version in : *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 29–38, 1987.*

[35] R. J. Vanderbei. *Linear Programming: Foundations and Extensions.* Kluwer Academic Publishers, Dordrecht, The Netherlands, second edition, 2001.

[36] R. J. Vanderbei, M. S. Meketon, and B. A. Freedman. A modification of Karmarkar's linear programming algorithm. *Algorithmica*, 1(4):395–407, 1986.

[37] H. Wolkowicz, R. Saigal, and L. Vandenberghe. *Handbook of semidefinite programming : theory, algorithms, and applications*, volume 27 of *International series in operations research and management science.* Kluwer Academic, Boston, 2000.

[38] S. J. Wright. *Primal–Dual Interior–Point Methods.* SIAM Publications, SIAM, Philadelphia, PA, USA, 1996.

[39] Y. Ye. An $O(n^3L)$ potential reduction algorithm for linear programming. *Mathematical Programming*, 50:239–258, 1991.

[40] Y. Ye. *Interior–Point Algorithms: Theory and Practice.* John Wiley & Sons, New York, USA, 1997.

[41] D. Yudin and A. Nemirovskii. Informational complexity and efficient methods for the solution of convex extremal problems. *Èkon.i Mat. Metody*, 12:357–369, 1976. (In Russian). Translated in: *Matekon 13(2)* 3-25.